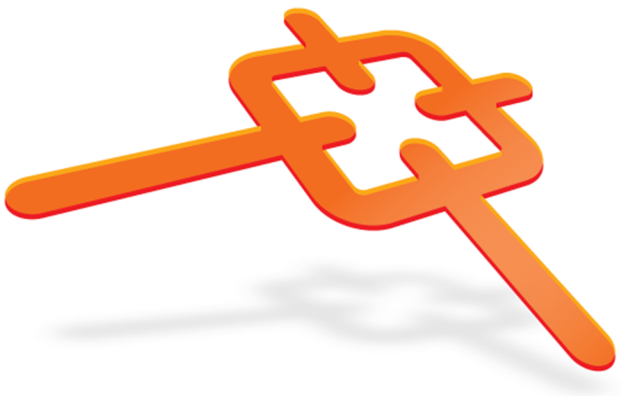
Portrait
Miner



Uplift Benutzerhandbuch

Version 7.0A

Pitney Bowes
Software

© 2013 Pitney Bowes Software Inc. Alle Rechte vorbehalten.

Dieses Dokument kann vertrauliche und eigentumsrechtlich geschützte Informationen enthalten, die Eigentum von Pitney Bowes Inc. bzw. seinen Tochter- und Beteiligungsgesellschaften sind.

Die Portrait Software, das Portrait Software-Logo und das Portrait-Zeichen der Portrait Software sind Warenzeichen der Portrait Software International Limited und dürfen in keiner Weise ohne die vorherige, ausdrückliche schriftliche Genehmigung von Portrait Software International Limited benutzt oder verwertet werden.

Warenzeichen

Die Namen von Firmen und Produkten, Marken, Logos und Zeichen, die hier genannt werden, sind möglicherweise Warenzeichen oder eingetragene Warenzeichen ihrer jeweiligen Eigentümer.

Portrait Software-Support

Sollten sie über den Inhalt dieser Dokumentation hinausgehende Hilfe benötigen, können sie in der Wissensbasis auf unserer Website **http://support.portraitsoftware.com** nachsehen und den Links zu ihrem Produkt folgen.

Sie können auch andere Portrait Miner-Dokumentationen herunterladen. Falls Sie keinen Benutzernamen und kein Kennwort haben oder sie diese vergessen haben sollten, kontaktieren Sie uns über einen der nachstehenden Kanäle.

Wenn sie Probleme bei der Nutzung, Installation oder Dokumentation von diesem Produkt feststellen, kontaktieren sie uns bitte auf einem der folgenden Wege:

E-Mail: support@portraitsoftware.com

Telefon

- USA/Kanada 1-800-335-3860 (gebührenfrei)
- Übrige Welt +44 800 840 0001

Bei einer Problemmeldung ist es hilfreich, wenn Sie uns Folgendes mitteilen können:

- Der Name der Software-Anwendung
- Die Umstände, unter denen das Problem auftrat
- Welche Fehlermeldungen Sie gegebenenfalls gesehen haben
- Die Version der von Ihnen benutzten Software

Pitney Bowes Software Inc.

July 01, 2013

Inhalt

Kapitel 1: Einleitung....9
Zweck....10
Zielgruppe....10
Überblick....10
Uplift-Ansatz....11
Schlüsselfunktionen....11

Kapitel 2: Portrait Uplift ausführen....13
Vorbereiten Ihrer Daten für Portrait Uplift....14
Portrait Uplift in Portrait Miner ausführen....14
Portrait Uplift von der Befehlszeile aus ausführen....18
Synopsis....18
Typische Verwendung....20
Portrait Uplift anhalten....23
Bagging-Uplift-Modellen auf neue Daten anwenden....23
Empfohlene Verzeichnisstruktur....23
SAS für das Aufrufen von Portrait Uplift konfigurieren....24
Portrait Uplift von SAS ausführen....24
Variablenauswahl aus SAS....25
Uplift-Modelle aus SAS heraus erstellen....26
Uplift-Modelle in SAS bewerten....28

Kapitel 3: Ansatzüberblick....29
Einleitung....30
Modellstruktur....30
Kampagnen-Design für Uplift-Analyse....31
Datenvorbereitung....32
Ansprechverzerrung und Modellvalidierung....32

Mehrfachansprachen.....35

Kapitel 4: Ansatz für Variablenauswahl und Klasseneinteilung.....37
Einleitung.....38
Einfluss auf Modellstabilität.....38
Automatisierte Variablenauswahl.....39
Automatisierte Klasseneinteilung.....40
Optimierte Klasseneinteilungen.....40
Wiederverwendung von Low Qini Estimates.....41
Ergebnisse und Berichte der Variablenauswahl.....42
Uplift Profile.....42
Zerlegungsvalidierung (der "Lila" Bericht).....43
Stabilitätsprofil (der "Trommelstock"-Bericht).....43

Kapitel 5: Modellerstellungsansatz.....45
Einleitung.....46
Stichproben und Stratifizierung.....46
Bagging.....47
Pruning.....48
Automatisierte Parameteruntersuchung.....48
Automatisierte Parameterempfehlung.....49
Endgültige Neuerstellung.....50
Modellverunreinigung.....50

Kapitel 6: Modellbewertungsansatz.....53
Einleitung.....54
Interpretation der Qini-Werte (Q und q0).....54
Interpretation von Qini-Kurven.....55
Eine Gains-Kurve für Uplift.....55
Vertikale Achse für Binäre Ergebnisse (pc pt).....55
Vertikale Achse für stetige Ergebnisse (photp).....55
Qini-Kurve als Diagnose.....56
Ergebnisse und Berichte.....57
Auswertungsprofile.....58
Zerlegungsvalidierung (der "Lila" Bericht).....58
Aufteilungsgewichtung.....59

Kapitel 7: Parameterrichtlinien....61
Einleitung....62
Pruning-Schwellenwert....62
Maximal Drei Versuche....63
Baum-Aufteilungsebenen....63
Mindestgrößen der Untersuchungsgesamtheit....63
Minimale Anzahl an Knoten pro Baum....64
Analysekandidaten und Klasseneinteilungen....64
Anzahl Bags....65
Anzahl Durchläufe....65
Stratifizierungsvariable....65

Kapitel 8: Parameter-Referenz....67
Einleitung....68
Focus-Beschreibungsparameter....69
focusName (erforderlich)....69
Modelltyp und Ausgabefelder: Überblick....69
modelType (erforderlich)....70
objectiveField (erforderlich)....70
outcomeSizeActual (optional)....70
outcomeSizeEstimate (für zweistufige Modelle erforderlich)....70
partitionField (erforderlich)....71
fixedTrainValidateField (optional)....71
stratificationField (optional, wird aber empfohlen)....71
championModelField (optional)....72
userWeightingField (optional)....72
yQiniAxis (optional, standardmäßiger kumulativer Uplift (pc pt))....73
Parameter zur Baumspezifizierung....73
minPopSize (optional; Standard 2000)....73
autoMinPopSize (optional; Standard 1)....73
searchMinPopSize (optional; Standard 0)....74
nLevels (optional; Standard 4)....74
Allgemeine Parameter....74
Parameter für die Variablenauswahl....74
setacsFocusName (optional)....74
allowOverwrite (optional; Standard 0)....75
nBestACs (optional; Standard: 5)....75

saveBestACs (optional; Standard: 1)....75
setacsNEvals (optional; Standard: 5)....75
setacsSigmaWeight (optional; Standard: Auto)....75
setacsRebin (optional; Standard: 0)....76
setacsRebinCriterion (optional; Standard: T)....76
setacsRebinMean (optional; Standard: Trim)....76
setacsUseRebinLQE (optional; Standard: 1)....77
Allgemeine Durchlaufparameter....77
nBagsValues (erforderlich)....77
pruneThresholdValues (erforderlich)....78
nRuns (optional; Standard: 1)....78
Pruning-Kontrollgruppenparameter....78
prune (optional; Standard: 1)....78
autoPrune (optional; Standard: 0)....79
pruneMethod (optional; Standard: sd)....79
minLeafNodes (optional; Standard: 4)....79
maxTreeAttempts (optional; Standard: 2)....79
propAnalysisCandidates (optional; Standard: 1.0)....79
Zufallszahlenerzeugung/Wiederholungsparameter ausführen....80
prngSetState (optional; standardmäßig leer)....80
prngCurrentState (optional)....80
prngInitialState (optional)....81
tvTrainProp (optional; Standard: 50%)....81
Berichts- und Ausgabeparameter....81
resultsDirStem (optional; Standard: Ergebnisse)....81
addTimeStamp (optional; Standard: 1)....81
saveFocusStem (optional; Standard: (leer))....82
saveModelStem (optional; Standard: Modell)....82
saveTreeStem (optional; Standard: (leer))....82
xmlResultsStem (optional; Standard: aus)....82
tvNewFieldName (optional; Standard: TrainValidate)....83
overallUpliftField (optional; Standard: UpliftScore)....83
ROI-Parameter....83
roiCurrencySymbol (optional; Standard: $)....83
roiCostOfTreatment (optional; Standard: 0.00)....83
roiValueOfBenefit (optional; Standard: 1.00)....83
roiCostOfFulfillment (optional; Standard: 0.00)....83
Parameter nur zur Information....83

currentNBags (festgelegt während der Modellerstellung.....83
currentPruneThreshold (festgelegt während der Modellerstellung).....84
applyFocusName.....84
Beispiel für eine vollständige Parameterdatei.....84

Kapitel 1

Einleitung

In diesem Abschnitt:

- Zweck 10
- Zielgruppe 10
- Überblick 10
- Uplift-Ansatz 11
- Schlüsselfunktionen 11

Zweck

Dieses Dokument beschreibt die Erstellung und Verwaltung von prädiktiven Modellen mit Portrait Uplift.

Zielgruppe

Das *Uplift Benutzerhandbuch* richtet sich an Statistiker und Marketing-Analysten mit Grundkenntnissen in:

- Prädiktives Modellieren
- Portrait Miner (Analytical Workbench)
- Marketinganalysen

Überblick

Portrait Uplift ist eine automatisierte Modellierungslösung, die über traditionelles prädiktives Modellieren von Kundenresonanz, Kundenrisiko oder Kundenbindung hinausgeht und modelliert, wie sich diese Wahrscheinlichkeit aufgrund der Marketing-Ansprache ändert.

Portrait Uplift arbeitet mit einem Marketingansatz, der auf zufällig ausgewählte Kundengruppen basiert, die im Rahmen einer Kampagne nicht angesprochen werden. Durch die Analyse nicht angesprochener und angesprochener Kunden ist Portrait Uplift in der Lage zu prognostizieren, wie die Verhaltenswahrscheinlichkeit eines jeden Kunden zunimmt (oder abnimmt), wenn er angesprochen wird.

In jedem Fall generiert Portrait Uplift vier verschiedene Resonanzsegmente. Zum Beispiel werden diese Segmente bei einer Cross-Selling-Kampagne wie folgt beschrieben:

Marktsegment	Beschreibung
Die Zugänglichen:	Kunden, die das Cross-Selling-Angebot nicht gekauft hätten, wenn sie nicht angesprochen worden wären.
Die Sicheren:	Kunden, die unabhängig davon, ob sie angesprochen werden oder nicht, kaufen.
Die hoffnungslosen Fälle:	Kunden, die unabhängig davon, ob sie angesprochen werden oder nicht, nicht kaufen.
Die Sensiblen:	Kunden, bei denen die Kampagne eine Negativresonanz auslöst.

Typische Werkzeuge für das prädiktive Modellieren zielen in erster Linie auf einen Kundenstamm, um zu ermitteln, welche Kunden kaufen würden und welche nicht. Portrait Uplift führt diesen Schritt tiefergehend aus und klassifiziert Kunden in die zuvor beschriebenen Segmente und ermöglicht Ihnen:

- Eine weitere Verfeinerung Ihrer Zielgruppe und Reduzierung der Gesamtkampagnenkosten
- Die Identifizierung des wichtigsten Marksegmentes: 'Die Zugänglichen'
- Die Identifizierung von Kunden, die negativ reagieren werden und somit möglicherweise für lange Zeit verloren gehen

Uplift-Ansatz

Portrait Uplift wurde um den Uplift-Baum-Algorithmus in Portrait Miner erstellt und enthält im Kern einen Best-Practice-Ansatz für die Erstellung von stabilen Uplift-Modellen. Dieser Ansatz basiert auf der praktischen jahrelangen Erfahrung führender Unternehmen der Finanzwirtschaft und Telekommunikation und schließt effektive Techniken zur Lärmreduzierung ein. Weiterhin identifiziert und isoliert er die kleinen, aber realen Uplift-Modelle vom zugrundliegenden Grundlinienverhalten.

Schlüsselfunktionen

Portrait Uplift liefert:

- Eine geführte visuelle Benutzeroberfläche für das Erstellen und Benutzen von Uplift-Modellen (siehe Abschnitt **Portrait Uplift in Portrait Miner ausführen** auf Seite 14), als auch eine automatisierte skriptfähige Befehlszeile als Schnittstelle (siehe Abschnitt **Portrait Uplift von der Befehlszeile aus ausführen** auf Seite 18)
- Eine automatisierte skriptfähige Befehlszeile als Schnittstelle für die Erstellung von Uplift-Modellen ohne Benutzerinteraktion (siehe Abschnitt **Portrait Uplift von der Befehlszeile aus ausführen** auf Seite 18)
- Eine Erweiterung für die SAS-Programmierumgebung, die es SAS-Anwendern ermöglicht, Uplift-Modelle direkt aus SAS-Programmen zu erstellen und zu benutzen, ohne dass neue Anwendungen erlernt werden müssen (siehe Abschnitt **Portrait Uplift von SAS ausführen** auf Seite 24)
- Modelle in SAS- oder SQL-Code ermöglichen die Bewertung von Uplift-Modellen direkt in diesen Umgebungen.
- Eine Anwendung entwickelt für transparente Resultate. Dies bedeutet eine hohe Transparenz der beitragenden Faktoren, Modell-Validierung und Uplift-Modell-Ansatz (siehe Abschnitte **Ansatzüberblick** auf Seite 29, **Ansatz für Variablenauswahl und Klasseneinteilung** auf Seite 37, **Modellerstellungsansatz** auf Seite 45)
- Für Microsoft Office optimierte Berichtsausgabe, um Benutzern schnelle Entscheidungen und Ad-hoc-Analysen zu ermöglichen. Diese Berichte enthalten die automatisierte Identifizierung der besten Kampagnen-"Entscheidungspunkte" wie maximaler Gewinn, minimale Ansprechkosten und maximale schrittweise zunehmende Resonanz (siehe Abschnitt **Modellbewertungsansatz** auf Seite 53).

Kapitel

2

Portrait Uplift ausführen

In diesem Abschnitt:

- Vorbereiten Ihrer Daten für Portrait Uplift 14
- Portrait Uplift in Portrait Miner ausführen 14
- Portrait Uplift von der Befehlszeile aus ausführen 18
- SAS für das Aufrufen von Portrait Uplift konfigurieren . .24
- Portrait Uplift von SAS ausführen 24

Vorbereiten Ihrer Daten für Portrait Uplift

Die Grundsequenz der Schritte für die Benutzung von Portrait Uplift wird nachfolgend, sowohl für interaktive Benutzung via Portrait Miner (siehe Abschnitt **Portrait Uplift in Portrait Miner ausführen** auf Seite 14) als auch in der Befehlszeile (siehe Abschnitt **Portrait Uplift von der Befehlszeile aus ausführen** auf Seite 18) beschrieben.

Beide Vorgehensweisen benutzen ähnliche Schritte. Portrait Miner führt den Benutzer mit einer Reihe von Assistenten Schritt für Schritt durch den Prozess, während die Befehlszeile eine genauere Kontrolle des Ansatzes und die Möglichkeit für eine automatisierte Batchverarbeitung bietet.

In beiden Fällen wird unser Basisfocus `example.ftr` genannt und die erste Aufgabe ist die Vorbereitung des Focus für die Analyse (mithilfe interaktiver Benutzung von Decision Studio oder Import von Metadaten), wobei besonders auf folgende Punkte geachtet wird:

- Festlegen einer entsprechenden Zielvariable, Zerlegungsfeld und Markierung der Analysekandidaten;
- Markierung von Feldern als kategorial, wenn erforderlich;
- Erstellen Sie optional eine festgelegte Trainings-/Validierungsaufteilung oder eine Stratifizierungsvariable, ein Gewichtungsfeld und/oder eine Topauswertung (siehe Abschnitt **Ansatzüberblick** auf Seite 29 für weitere Informationen).
- Fügen Sie den Analysekandidaten optional entsprechende Klasseneinteilungen hinzu, falls Sie keine Optimierung der Klasseneinteilungen planen (siehe Abschnitt **Ansatz für Variablenauswahl und Klasseneinteilung** auf Seite 37);

Portrait Uplift in Portrait Miner ausführen

Um Portrait Uplift in Portrait Miner zu benutzen, klicken Sie mit der rechten Maustaste auf das Modellierungsdatenset (oder Parameterdatei), um einen der nachfolgend beschriebenen Assistenten aufzurufen.

Hinweis:

Im Portrait Uplift Installationsordner befindet sich eine `params.xml`-Datei. Diese stellt websiteweite Standards für Variablenauswahl und Modellerstellungsparameter bereit (siehe Abschnitt **Parameter-Referenz** auf Seite 67). Sie wird verwendet, wenn Sie einen dieser Assistenten zum ersten Mal aufrufen. Die von Ihnen verwendeten Parameter werden gespeichert, nachdem Sie einen dieser Assistenten zum ersten Mal benutzt haben. Sie werden bei der nächsten Ausführung des Assistenten wieder geladen. Hierdurch können Sie Ihre eigenen spezifischen Standardparameter für ein bestimmtes Datenset erstellen. Sie können die Parameterdatei auch direkt bearbeiten, um Parameter zu verändern, die nicht vom Assistenten angezeigt werden. Jede Änderung wird mit allen Einstellungen zusammengeführt, die Sie später wiederum im Assistenten bearbeiten können.

- Bereiten Sie Ihren Focus (`example.ftr`) wie zuvor beschrieben vor.

- Um das Datenset für das Uplift-Modellierung zu beurteilen, klicken Sie optional mit der rechten Maustaste auf den Focus (`example.ftr`) und wählen **Portrait Uplift** > **Daten bewerten**. Optional können Sie entsprechende Werte für die ROI-Parameter im Dialogfeld **Portrait Uplift - Daten bewerten** festlegen.

  Wenn Sie auf **Start** klicken wird ein Bericht über die Daten und ihre Eignung für die Uplift-Modellierung in der Registerkarte **Ausgabe** angezeigt.

- Optional, um die besten Analysekandidaten auszuwählen, klicken Sie mit der rechten Maustaste auf den Focus, den Sie aus (`example.ftr`) auswählen möchten und wählen **Portrait Uplift** > **Variablen auswählen** (Abbildung 1). Der Ansatz zur Variablenauswahl wird in Abschnitt **Ansatz für Variablenauswahl und Klasseneinteilung** auf Seite 37 näher beschrieben.

  Wählen Sie im Dialogfeld **Portrait Uplift - Variablen auswählen** (Abbildung 2), ob Sie die Klasseneinteilungen optimieren möchten (und, in diesem Fall, die maximal zu berücksichtigende Anzahl an Klassen), die Anzahl der auszuwählenden endgültigen Analysekandidaten, ob Sie ein Fallgewichtungsfeld zur Ansprechverzerrung verwenden möchten (siehe Abschnitt **userWeightingField (optional)** auf Seite 72) und wählen Sie einen Speicherort für den Ausgangsfocus aus (der nicht mit dem für den Eingangsfocus übereinstimmen muss). Die Parameterdatei wird automatisch mit einem Dateinamen basierend auf den Eingangs- und Ausgangsfoci gespeichert.

  Klicken Sie auf **Start**.

  Die resultierende Ausgabe wird in einem Ordner unter einem Namen ähnlich `resultsJJJJMMTT-HHMMSS` gespeichert, wobei `JJJJMMTT` ein achtstelliges Datum und `HHMMSS` ein Zeitstempel ist, z.B. `results20121212-121212`. Falls Sie sich für die Optimierung der Klasseneinteilungen entscheiden werden in `uplift-optimizer-setacs-report_JJJJMMTT-HHMMSS.xsl` eine Reihe optimierter Klasseneinteilungsdiagramme enthalten sein.

- Damit Portrait Uplift die "besten" Modellparameter finden kann, klicken Sie mit der rechten Maustaste auf den Focus, den Sie benutzen möchten (in der Regel der Ausgangsfocus der Variablenauswahl `example_BestACs.ftr`) und wählen **Portrait Uplift** > **Modelle erstellen und analysieren**. Der für die Modellerstellung angewendete Ansatz wird im Abschnitt **Modellerstellungsansatz** auf Seite 45 näher beschrieben.

  Wählen Sie im Dialogfeld **Portrait Uplift - Modelle erstellen und analysieren** die für den Erstellungsprozess erforderlichen Parameter aus (befolgen Sie dabei die Anweisungen in Abschnitt **Parameterrichtlinien** auf Seite 61):

  - Die pro Modell zu verwendende Anzahl von Bäumen (`nBagsValues`, Abschnitt **nBagsValues (erforderlich)** auf Seite 77);
  - Die zu erstellende Anzahl von Uplift-Modellen für jeden Parametersatz (`nRuns`, Abschnitt **nRuns (optional; Standard: 1)** auf Seite 78);
  - Ob mehrere Mindestsegmentgrößen analysiert werden sollen (`searchMinPopSize`, Abschnitt **searchMinPopSize (optional; Standard 0)** auf Seite 74);
  - Ob der Software erlaubt werden soll, die Mindestgröße der Untersuchungsgesamtheit festzulegen (`autoMinPopSize`, Abschnitt **autoMinPopSize (optional; Standard 1)** auf Seite 73);
  - Legen Sie die Mindestgröße der Untersuchungsgesamtheit eines Entscheidungsbaumsegments (`minPopSize`, Abschnitt **minPopSize (optional; Standard 2000)** auf Seite 73) fest;

- Die Mindestanzahl von Blattsegmenten nach Pruning (`minLeafNodes`, Abschnitt **minLeafNodes (optional; Standard: 4)** auf Seite 79);
- Die maximale Ebenenanzahl für das Erstellen eines Entscheidungsbaums (`nLevels`, Abschnitt **nLevels (optional; Standard 4)** auf Seite 74);
- Legen Sie den Namen der Ausgangsparameterdatei (`nLevels`, Abschnitt **nLevels (optional; Standard 4)** auf Seite 74) fest.

Klicken Sie auf **Weiter** um sich die Registerkarte **Finanzen** anzusehen und legen Sie (optional) die Finanzparameter fest.

Klicken Sie auf **Weiter**, um sich die Registerkarte **Erweitert** anzusehen und bestimmen Sie (optional):

- Den zu erstellenden Modelltyp (`modelType`, Abschnitt **modelType (erforderlich)** auf Seite 70);
- Ob die Software den Pruning-Schwellenwert auswählen soll;
- Eine Liste der zu verwendenden Pruning-Schwellenwerte (`pruneThresholdValues`, Abschnitt **pruneThresholdValues (erforderlich)** auf Seite 78);
- Ob Sie einen bestimmten Startwert verwenden, statt diesen von Portrait Uplift auswählen zu lassen (`prngCurrentState`, Abschnitt **prngCurrentState (optional)** auf Seite 80);
- Das für Training/Bewertung zu verwendende Feld, falls überhaupt (`fixedTrainValidateField`, Abschnitt **fixedTrainValidateField (optional)** auf Seite 71);
- Den für das Trainingsset zu verwendenden Datenanteil, falls das Training-/Bewertungsfeld automatisch erstellt werden soll (`tvTrainProp`, Abschnitt **tvTrainProp (optional; Standard: 50%)** auf Seite 81);
- Das für die Gewichtung zu verwendende Feld, falls Datensätze unterschiedliche Anzahlen von tatsächlichen Kunden repräsentieren, (`userWeightingField`, Abschnitt **userWeightingField (optional)** auf Seite 72);
- Den Anteil von zufällig ausgewählten Analysekandidaten die verwendet werden sollen (`propAnalysisCandidates`, Abschnitt **propAnalysisCandidates (optional; Standard: 1.0)** auf Seite 79);
- Die maximale Anzahl Versuche für jede Baumerstellung (`maxTreeAttempts`, Abschnitt **maxTreeAttempts (optional; Standard: 2)** auf Seite 79);

Klicken Sie auf **Weiter**, um sich die Registerkarte **Ergebnisparameter** anzusehen und bestimmen Sie (optional):

- Den Namen des zu erstellenden, vorhergesagten Uplift-Feldes (`overallUpliftField`, Abschnitt **overallUpliftField (optional; Standard: UpliftScore)** auf Seite 83);
- Ob die Zwischenfoci nach jedem erstellten Modell gespeichert werden sollen (`saveFocusStem`, Abschnitt **saveFocusStem (optional; Standard: (leer))** auf Seite 82);
- Das für ein Top-Modell zu verwendende Feld, falls überhaupt (`championModelField`, Abschnitt **championModelField (optional)** auf Seite 72);
- Ob ein SAS-Modell erstellt werden soll.

Klicken Sie auf **Start**.

Hierdurch wird eine Ausgabe in einem Ordner mit einem Namen ähnlich `resultsJJJJMMTT-HHMMSS` erstellt, wobei `JJJJMMTT` ein achtstelliges Datum und `HHMMSS` ein Zeitstempel ist, z.B. `results20121212-121212` (Abbildung **9** im Abschnitt **Modellerstellungsansatz** auf Seite 45).

Während der Ausführung des Vorganges werden in der Registerkarte **Ausgabe** Informationen angezeigt. Fehler und Warnungen werden in der Regel in einer anderen Registerkarte angezeigt. Klicken Sie nach Beendigung des Vorgangs auf **Weiter**, um die Ergebnisse zu erstellen: In dem oben genannten Ordner wird ein Excel-Arbeitsblatt erstellt, welches automatisch geöffnet wird.

Das Excel-Arbeitsblatt kann auch zu einem späteren Zeitpunkt geöffnet werden, indem Sie zum Ordner navigieren, mit der rechten Maustaste auf die Datei klicken und **An Excel senden** auswählen. Die Arbeitsblattdatei wird unter einem Namen ähnlich "uplift-optimizer-report_JJJJMMTT-HHMMSS.xls" gespeichert. Vererbte HTML "BestACs_recommended_parameters.xml" Berichte werden ebenfalls erstellt. Die Berichte werden näher im Abschnitt **Modellbewertungsansatz** auf Seite 53 beschrieben.

Falls ein Parameterset empfohlen wurde, wird zusammen mit der Focusdatei eine neue Parameterdatei mit einem Namen ähnlich *<focusname>*_recommended_parameters.xml (`example_BestACs_recommended_params Deploy Recommended Model .xml`) erstellt, wobei die Empfehlung im Arbeitsblatt hervorgehoben wird. Die neue Parameterdatei enthält Parameter, die für die nächste Stufe der Erstellung des endgültigen Modells geeignet sind.

- Alternativ können die Analyse und die Parameterauswahl für die Erstellung des endgültigen Modells "nach Augenmaß" durchgeführt werden. Wählen Sie die besten Modellparameter aus, indem Sie besonderes auf die Validierungsleistung (das heißt, wie eng beieinander die Trainings- und Validierungslinien im Diagramm liegen) und den allgemeinen (Qini) Wert q_0, den Verlauf der Kurve, den maximal (oder minimal) erreichten Uplift, den Anteil der Zieluntersuchungsgesamtheit usw. achten. Erstellen Sie, nachdem Sie einen passenden Parametersatz für die Modellerstellung gefunden haben, basierend auf dem gesamten Datenset (ohne Trainings-/Validierungsaufteilung) ein endgültiges Bagging für die Modellerstellung. Weitere Informationen finden Sie im Abschnitt **Endgültige Neuerstellung** auf Seite 50.
- Um Portrait Uplift erneut zu starten und das empfohlene Modell zu erstellen, klicken Sie mit der rechten Maustaste auf den Focus (`example_BestACs.ftr`) oder auf die empfohlene Parameterdatei und wählen **Portrait Uplift** > **Empfohlenes Modell erstellen**. Dadurch wird einzelnes Modell mit den im vorigen Schritt empfohlenen Parametern erstellt, jedoch mit einer großen Anzahl Bags und ohne Validierungsdaten.

  Wählen Sie im Dialogfeld **Portrait Uplift - Empfohlenes Modell erstellen** die Parameter für den Erstellungsprozess aus und klicken Sie auf **Start**.

  Während der Ausführung des Vorganges werden in der Registerkarte **Ausgabe** Informationen angezeigt. Nach Abschluss des Vorganges werden neue Ergebnisberichte erstellt und das Modell wird gemeinsam mit dem Focus in einer Datei mit einem Namen ähnlich `<focusname>_recommended_model.xml`, (`BestACs_recommended_model.xml`) erstellt. Diese Modelldatei ähnelt dem Modell, das im Ergebnisordner erstellt wurde und wurde derart bearbeitet, dass bei Anwendung des Modells nur ein einziges Feld erzeugt wird.
- Um das endgültige Modell in einem neuen Datenset anzuwenden, klicken Sie mit der rechten Maustaste auf die im vorigen Schritt erstellt Modelldatei (`example_bestACs_recommended_model.xml`) oder auf die Focusdatei, die Sie anwenden möchten und wählen **Portrait Uplift** > **Modell anwenden**.

  Wählen Sie im Dialogfeld **Portrait Uplift - Modell anwenden** die Parameter aus und klicken Sie auf **Start**. (Diese Aktion ähnelt im Wesentlichen der Portrait Miner-Aktion 'Felder ableiten'.)
- Zu Anhalten eines laufenden Vorgangs klicken Sie in Portrait Miner mit der rechten Maustaste auf den Ergebnisordner oder auf eine XML-Datei im Ergebnisordner und wählen **Portrait Uplift** > **Anhalten**.

- Während sich Ergebnisordner ansammeln, möchten Sie evtl. einige davon löschen. Um einen einzelnen Ordner aus Portrait Miner zu entfernen, klicken Sie mit der rechten Maustaste auf eine beliebige `.ftr-` oder `.xml-`Datei im Ordner und wählen **Portrait Uplift** > **Ergebnisse löschen**. Um einen oder mehrere Ergebnisordner aus Portrait Miner zu löschen, klicken Sie mit der rechten Maustaste auf eine beliebige `.ftr-` oder `.xml-`Datei im Ergebnisordner und wählen **Portrait Uplift** > **Ergebnisse löschen**.

  Wählen Sie im Dialogfeld **Portrait Uplift - Ergebnisse löschen** die Ergebnisordner aus, die Sie löschen möchten.

- Portrait Uplift erstellt automatisch Modellbeschreibungen sowohl im Portrait Miner-Format als auch im SAS-Code (sofern ausgewählt). Um ein Modell manuell in SAS-Format zu konvertieren klicken Sie mit der rechten Maustaste auf die XML-Modelldatei (im Unterordner des Ergebnisordners namens `Modelle`) und wählen **Konvertieren** > **in SAS**.

  Das Modell wird in ein SAS DATA-Schrittfragment konvertiert. In der Regel wird der Code zwischen einem DATA-Statement, das sich auf ein Eingabedatenset mit entsprechenden Quellenvariablen bezieht (der erstellte Code listet erforderliche Eingabevariablen in einem Eröffnungskommentar auf) und dem nachfolgenden RUN-Statement eingefügt. Beispiel:

```
data scores;
   set inputs;
   <...paste converted model here...>
run;
```

  Weitere Informationen über den SAS DATA-Schritt finden Sie in der SAS-Dokumentation.

- Um ein Modell manuell in SQL-Format zu konvertieren klicken Sie mit der rechten Maustaste auf die XML-Modelldatei (im Unterordner des Ergebnisordners namens `Modelle`) und wählen **Konvertieren** > **in SQL**.

Portrait Uplift von der Befehlszeile aus ausführen

Synopsis

Ein Großteil der Funktionalität von Portrait Uplift wird mithilfe des Befehls `qsupliftplus` aufgerufen. Es gibt drei Basisvarianten dieses Befehls:

- `qsupliftplus -s [ params.xml ]`: wird zur Auswahl von Vorhersagevariablen (Analysekandidaten) verwendet, die für Portrait Uplift genutzt werden. Hierfür werden für jeden Analysekandidaten im Eingabefocus Low Qini Estimates und optional eine Stratifizierungsvariable erstellt.
- `qsupliftplus [ params.xml ]`: führt eine Reihe an Modellerstellungen aus, von denen jede einen Bagging-K-Way-Uplift-Baum produziert und die Ergebnisse an eine Trainings-/Validierungsaufteilung meldet.
- `qsupliftplus -a [ params.xml ]`: wendet mit Portrait Uplift erstellte Modelle auf einen neuen Focus an und speichert Informationen aus der ursprünglichen Modellerstellung zusammen mit Qini-Informationen über die angewendeten Modelle, um Qini-Diagramme produzieren zu können (siehe Abschnitt **Bagging-Uplift-Modellen auf neue Daten anwenden** auf Seite 23).

Der Befehl `qsupliftplus` unterstützt außerdem folgende Argumente:

Option	Effekt
-h	Hilfenachricht anzeigen
-v	Versionsinformation anzeigen
-d	Debugging-Ausgabe aktivieren
-D	Gesamte Stapelverfolgung bei Fehler verwerfen
-o <*params datei*>	Eine XML-Parameterdatei verwerfen. Falls der optionale Focusname ebenfalls vorhanden ist werden daraus relevante Parameter geschlussfolgert.
-P <*Zerlegung feld*>	Spezifiziert das Zerlegungsfeld wenn eine Parameterdatei mit der Option -o erstellt wird
-O <*zielvariable feld*>	Bestimmt das Feld der Zielvariable wenn eine Parameterdatei mit der Option -o erstellt wird

Die detaillierten Ausführungsoptionen aller `qsupliftplus`-Varianten werden durch eine XML-Parameterdatei gesteuert. Standardmäßig wird im aktuellen Verzeichnis nach einer Datei mit dem Namen `params.xml` gesucht. Es kann jedoch eine alternative Parameterdatei verwendet werden, indem ihr Name in der Befehlszeile eingegeben wird, z.B. `qsupliftplus -s /some/path/myconfig.xml`.

Eine Parameterdateivorlage kann erstellt werden, indem eine der oben genannten Basisvarianten ausgeführt wird, ohne dass dabei weitere Parameter bestimmt werden, oder indem lediglich der Pfadname zu einem Eingabefocus angegeben wird (in diesem Fall werden einige der Parameter aus diesem Focus geschlussfolgert).

Eine komplette Übersicht über die Parameter finden Sie im Abschnitt **Parameter-Referenz** auf Seite 67. Nachfolgend werden am häufigsten verwendeten Parameter zusammengefasst:

Tabelle 1:

Focusbeschreibung	Variablenauswahl
`setacsFocusName`: Focus für Variablenauswahl	`nBestACs`: für auszuwählende Variablenauswahl
`focusName`: wird nach der Variablenauswahl geschrieben und bei der Modellerstellung gelesen	`setacsRebin`: 1, 0, *oder* `binfile.xml`
`modelType: auto, binary, real`	`setacsRebinCriterion`: `L`, `M`, `E`, *oder* `T`
`objectiveField`: Ergebnis	

Focusbeschreibung	Variablenauswahl
`partitionField`: Experimentalgruppe=1, Kontrollgruppe=0 `stratificationField`: Zusammenfassung von Schlüsselfeldern `championModelField`: In Diagrammen dargestellt `allowOverwrite`: Erlaubt das Überschreiben nach der Variablenauswahl	

Baumspezifikation	General
`searchMinPopSize`: 0 oder 1 `autoMinPopSize`: 0 oder 1 `minPopSize`: c. 500-4 000 `nLevels`: von Aufteilungen `maxTreeAttempts`: um minLeafNodes zu genügen `tvTrainProp`: c. 50–70%; dann 100% `fixedTrainValidateField`: String-Werte "T" und "V" `propAnalysisCandidates`: wird für jeden Baum verwendet `userWeightingField`: Fallgewichtung für jeden Datensatz	`nBagsValues`: c. 10–20 `pruneThresholdValues`: z.B. [3%, 2%, 1%, 0,5%] `nRuns`: pro oben genanntem Wert `minLeafNodes`: c. 3–8 (nach Pruning)

Ergebnisse aus der Ausführung von Portrait Uplift werden in eine XML-Datei geschrieben, aus welcher die Diagramme und Berichte produziert werden können. Verwenden Sie den Befehl `qsupliftresults`, um die Ergebnisse im aktuellen Verzeichnis zu verarbeiten oder bestimmen Sie einen optionalen Verzeichnisnamen, um die Ergebnisse dort zu verarbeiten.

Typische Verwendung

Die Basissequenz der Operationen für eine typische Verwendung von Portrait Uplift wird nachfolgend beschrieben. In diesem Fall nehmen wir an, dass unser Basisfocus `example.ftr` heißt und, wie beschrieben, vorbereitet wurde.

1. Erstellen Sie eine Parameterdateivorlage, falls Sie nicht schon über eine passende verfügen. Wenn Sie eine Optimierung der Klasseneinteilungen durchführen und/oder Analysekandidaten auswählen möchten, tun Sie dies durch Ausführung von

   ```
   qsupliftplus -s focus.ftr
   ```

   Wenn Sie Portrait Uplift ohne Variablenauswahl ausführen möchten, verwenden Sie stattdessen

   ```
   qsupliftplus focus.ftr
   ```

   (Der Hauptunterschied zwischen Beiden ist, dass wenn `-s` verwendet wird, die Variablenauswahl auf den vorgegebenen Focus erfolgt und danach die Eingabe für die Modellerstellung erzeugt wird. Wenn nur die Modellerstellung ausgeführt wird, wird der vorgegebene Focus direkt verwendet.)

   Dadurch werden alle Parameter, einschließlich der Einstellung eines angemessenen `minPopSize`, so genau wie möglich ausgefüllt, falls die Zielvariable und Zerlegungsfelder im festgelegten Focus bestimmt wurden.

   Wenn die Ausgabedatei einen anderen Namen als `params.xml` haben soll, können Sie diesen festlegen, indem Sie die Flag `-o` verwenden. Z.B. wird

   ```
   qsupliftplus -s -o foo.xml focus.ftr
   ```

   eine Parameterdateivorlage unter dem Namen `foo.xml` erstellen.

   Sie können den Focusnamen weglassen. In diesem Fall wird eine generische Parameterdateivorlage erstellt.

2. Bearbeiten Sie die Parameterdatei `params.xml` und speichern Sie sie für gewöhnlich unter einem neuen Namen. In diesem Fall wäre `exampleParams.xml` angemessen. Weitere Informationen über den Inhalt der Datei finden Sie im Abschnitt **Parameter-Referenz** auf Seite 67. Mitunter ist es hilfreich die Parameter in einem Unterverzeichnis mit dem Namen `params` zu speichern, um ein Durcheinander zu vermeiden.
3. Optional können Sie die besten Analysekandidaten auswählen und die Klasseneinteilungen optimieren (siehe Abschnitt **Ansatz für Variablenauswahl und Klasseneinteilung** auf Seite 37). Verwenden Sie dazu einen Befehl wie

   ```
   qsupliftplus -s params/exampleParams.xml
   ```

   (Das setzt voraus, dass die Parameterdatei in einem Unterverzeichnis params gespeichert wurde.)

   Ist die Variablenauswahl einmal abgeschlossen, erstellen Sie den Ergebnisbericht, indem Sie in das erstellte Unterverzeichnis wechseln und den Befehl `qsupliftresults` verwenden, z.B.

   ```
   cd results20121212-121212
   qsupliftresults
   ```

4. Optional können Sie eine Stratifizierungsvariable (z.B. `strat`) zur Verwendung durch Portrait Uplift, wie im Abschnitt **Stichproben und Stratifizierung** auf Seite 46 beschrieben, erstellen und die Parameterdatei `exampleParams.xml` entsprechend bearbeiten. (Dieser Schritt wird normalerweise automatisch während der Variablenauswahl ausgeführt.)
5. Starten Sie das Erstellen von Modellen durch Portrait Uplift (siehe Abschnitt **Modellerstellungsansatz** auf Seite 45) durch Eingabe von:

   ```
   qsupliftplus params/exampleParams.xml
   ```

Damit wird eine Ausgabe in einem Verzeichnis mit einem Namen ähnlich `resultsJJJJMMTT-HHMMSS` erstellt, wobei `JJJJMMTT` ein achtstelliges Datum und `HHMMSS` ein Zeitstempel, z.B. `results20121212-121212`, ist.

6. Erstellen Sie jetzt die Qini-Diagramme und den Modellbericht (siehe Abschnitt **Modellbewertungsansatz** auf Seite 53), indem Sie in das Ergebnisverzeichnis wechseln und folgendes eingeben:

```
qsupliftresults
```

   Dadurch wird eine PDF-Datei der Diagramme mit einem Namen ähnlich `results20121212-121212.pdf` und eine Webseite mit einer Bezeichnung ähnlich `report20121212-121212.html` erstellt. Diese verweist auf die Diagramme und andere zusammengefasste Informationen der Durchläufe.

7. Wiederholen Sie dieses einige Male, um bessere Parameter für die Modellerstellung zu finden und indem Sie möglicherweise einige Felder oder ihre Klasseneinteilungen anpassen. Einige der wichtigeren Parameter für die Modellerstellung sind:

   - Der Pruning-Schwellenwert, gesteuert durch `pruneThresholdValues`;
   - Die Mindestgröße der Untersuchungsgesamtheit `minPopSize`;
   - Die minimale Anzahl an Blattverzweigungspunkten pro Baum `minLeafNodes`.

   Diese drei Parameter interagieren intensiv miteinander, sodass Änderungen an einem Parameter auch Änderungen der anderen erfordern können. Natürlich gibt es zahlreiche weitere Parameter, die unter gewissen Bedingungen auch angepasst werden sollten. Weitere Informationen finden Sie im Abschnitt **Parameterrichtlinien** auf Seite 61.

8. Legen Sie bei der Bewertung der Modellleistung besonderes Augenmerk auf die Validierungsleistung (das heißt, wie nah liegen die Trainings- und Validierungslinien im Diagramm nebeneinander). Achten Sie aber auch auf den allgemeinen (Qini) Wert q_0, den Verlauf der Kurve, dem maximal (oder minimal) erreichten Uplift, das Verhältnis der Zieluntersuchungsgesamtheit, usw. Bedenken Sie, dass Qini-Kurven einen identischen Qini-Wert, aber sehr unterschiedliche Formen haben können. Welches Modell am besten geeignet ist, hängt von Ihrer Zielstrategie und Ihren Zielkriterien ab.
9. Nachdem Sie einen passenden Parametersatz für die Modellerstellung gefunden haben, berücksichtigen Sie einige endgültige Bagging-Modelle unter Verwendung des gesamten Datensets (keine Trainings-/Validierungsaufteilung). Weitere Informationen finden Sie im Abschnitt **Endgültige Neuerstellung** auf Seite 50.
10. Wenn Sie mit einem Modell zufrieden sind, möchten Sie dieses vielleicht zur Bewertung der passenden Untersuchungsgesamtheit verwenden. Das wird häufig mit Hilfe von `qsderive` durch einen Befehl erreicht, wie

```
qsderive -derivations modelB03T2,00pcR0001.xml
       -input deploy.ftr -output deployScored.ftr
```

   wobei `deploy.ftr` der zu bewertende Focus ist.

11. Später können Sie dem angewendeten Focus tatsächliche Ergebnisse hinzufügen und dann für die tatsächliche Modellleistung in einem Qini-Diagramm darstellen. Dies können Sie mit dem Befehl `qsupliftplus -a` durchzuführen. Typischerweise mit einer Sequenz wie:

```
cd results20121212-121212
qsupliftplus -a out.xml ../foci/deployScoredplusActual.ftr
```

Siehe Abschnitt **Bagging-Uplift-Modellen auf neue Daten anwenden** auf Seite 23 für weitere Informationen.

Portrait Uplift anhalten

Aus verschiedenen Gründen ist es normalerweise keine gute Idee, Portrait Uplift anzuhalten, indem man diesen mit `control-C` oder ähnlich abbricht. Neben anderen unerfreulichen Konsequenzen kann dies dazu führen, dass Foci "gesperrt" bleiben und somit schwer zu verwenden sind.

Eine saubere Methode, um dieses zu verhindern, ist eine Datei mit dem Namen `stop.now` im Ergebnisverzeichnis zu erzeugen. Diese veranlasst die Software zum nächsten sicheren Zeitpunkt anzuhalten. Das geschieht in der Regel recht schnell.

Bei UNIX-Systemen ist der einfachste Weg hierfür der Befehl

```
touch stop.now
```

im Ergebnisverzeichnis.

Bei Windows-Systemen ist der einfachste Weg wahrscheinlich das Kopieren und Einfügen von `params.xml`, die bereits vorhanden sein sollte, und das anschließende Umbenennen von `Copy of params.xml` in `stop.now`.

Bagging-Uplift-Modellen auf neue Daten anwenden

Ebenso ist bei Portrait Uplift der Befehl `qsupliftplus -a` verfügbar. Er kann verwendet werden, um eine Reihe von zuvor mit Portrait Uplift erstellte Uplift-Modelle auf neue Daten anzuwenden.

Alle Schlüsselinformationen über einen Durchlauf werden von Portrait Uplift in der durch den Parameter `xmlResultsStem` bestimmten Datei gespeichert. Da die Standardeinstellung hierfür `out` ist, wird diese Datei normalerweise als `out.xml` bezeichnet.

Der Befehl `qsupliftplus -a` liest die XML-Ergebnisdatei `out.xml` und schreibt, basierend auf den darin enthaltenen Informationen eine neue XML-Ergebnisdatei in ein Unterverzeichnis, das nicht nur die Qini-Diagramme für die ursprünglichen Trainings- und Validierungsdaten, sondern auch eine Zeile für die neuen ("angewendeten") Daten enthält.

Nehmen wir zum Beispiel an, dass Portrait Uplift in einem Verzeichnis mit dem Namen `Projects` ausgeführt wird und zunächst eine Ausgabe im Ergebnisverzeichnis `results20121212-121212` erzeugt und Foci aus dem Verzeichnis `foci` verwendet. Am folgenden Tag um 13:13 Uhr und 13 Sekunden wird, ausgehend vom Verzeichnis `Projects`, folgende Befehlssequenz erstellt:

```
cd results20121212-121212
qsupliftplus -a out.xml ../foci/newdata.ftr
cd results20051213-131313
```

Empfohlene Verzeichnisstruktur.

Alle Datensets, Parameter, Ergebnisse (und mehr) im gleichen Verzeichnis zu haben kann verwirrend sein, besonders wenn mehrere Durchläufe erfolgen.

Einige Personen ziehen es vor eine Verzeichnisstruktur zu verwenden, um die Daten zu ordnen.

Bei einem ProjectX genannten Projekt mit einem Basisverzeichnis `ProjectX` kann es hilfreich sein die Unterverzeichnisse `ProjectX/foci`, `ProjectX/params` und `ProjectX/runs` zu erstellen. Falls Sie dies tun, aber immer noch Befehle aus dem Verzeichnis `ProjectX` ausgeben, müssen Sie den Parameter `resultsDir` in `runs/results` ändern.

Obwohl Portrait Uplift das Verzeichnis `ProjectX/runs` bei dieser Parametereinstellung nicht für Sie erstellt, wird es die Ergebnis-Unterverzeichnisse korrekt als `ProjectX/runs/results20121212-121212` erstellen,

und so weiter. Ähnlich ist es, wenn Sie Ihre Parameterdatei mit Hilfe eines Befehls wie

```
qsupliftplus -s foci/CampaignX.ftr
```

erstellen. Dabei werden die Parameter korrekt in der Parameterdatei festgelegt, um eine Durchführung vom Verzeichnis `ProjectX` aus zu ermöglichen.

SAS für das Aufrufen von Portrait Uplift konfigurieren

Mit Portrait Uplift wird ein experimenteller Satz SAS-Makros zur Verfügung gestellt, damit die Hauptoperationen direkt von einem SAS-Programm aus ausgeführt werden können. Um diese Makros verwenden zu können müssen Sie Ihre SAS-Installation so konfigurieren, dass sie die Datei `autoexec` in `<`*pmhome*`>/ext/upliftplus/autoexec.sas` lädt (wobei `<`*pmhome*`>` das Portrait Miner Installationsverzeichnis ist).

Eine Möglichkeit, um diese `autoexec`-Datei verwenden zu können ist das Konfigurieren der SAS Desktop-Verknüpfung. Zum Beispiel:

```
"C:\Program FilesSASSAS Learning Edition 4.1SAS.exe"
  -autoexec "PMHOME\ext\upliftplus\autoexec.sas"
```

Portrait Uplift von SAS ausführen

Mit Portrait Uplift wird ein experimenteller Satz SAS-Makros zur Verfügung gestellt, damit die Hauptoperationen direkt von einem SAS-Programm aus ausgeführt werden können.

Wenn SAS nicht entsprechend konfiguriert wurde um die Portrait Uplift-Makros automatisch zu laden, fügen Sie Zeilen in Ihren SAS-Code ein, die folgendermaßen aussehen können, wobei `c:/PortraitMiner` der Installationsordner von Portrait Uplift ist:

```
libname uplift "c:\PortraitMiner\ext\upliftplus" ;
options mstored sasmstore=uplift ;
```

Variablenauswahl aus SAS

Ein Makro `uplift_variable_selection` ermöglicht den Zugriff auf die Funktionalität der Variablenauswahl innerhalb von Portrait Uplift. Dazu werden folgende Parameter benötigt:

data (Erforderlich: Dateipfad) Speicherort des SAS-Datensets (`.sas7bdat`) für die Analyse.

objectiveField (Erforderlich: Variablenname) Name der abhängigen Variable für die Analyse: Siehe Abschnitt **objectiveField (erforderlich)** auf Seite 70.

partitionField (Erforderlich: Variablenname) Name der Experimental-/Kontrollgruppenaufteilungsvariable: Siehe Abschnitt **partitionField (erforderlich)** auf Seite 71.

analysisFields (optional: durch Komma getrennte Liste der Variablennamen) Liste der für die Variablenauswahl zu berücksichtigenden unabhängigen Variablen. Wenn dieser nicht vorhanden ist, werden alle Variablen berücksichtigt.

xanalysisFields (optional: durch Komma getrennte Liste der Variablennamen) Liste der für die Variablenauswahl auszuschließenden unabhängigen Variablen.

categoricalFields (optional: durch Komma getrennte Liste der Variablennamen) Liste der Variablen, die als nominale Variablen interpretiert werden sollen.

nBestACs (Erforderlich: Integer) Anzahl an auszuwählenden Variablen: Siehe Abschnitt **nBestACs (optional; Standard: 5)** auf Seite 75.

setacsRebin (optional: 0 oder 1) Wenn auf 1 gesetzt, wird die variable Klasseneinteilung basierend auf einer Wechselwirkung mit dem Uplift optimiert: Siehe Abschnitt **setacsRebin (optional; Standard: 0)** auf Seite 76.

setacsRebinMaxBins (optional: Integer) Maximale Anzahl von Klassen (Behälter), die bei der Optimierung der Variablen-Klassifizierung berücksichtigt werden soll.

stratificationField (optional: Variablenname) Variable, die zur Stratifizierung der Daten verwendet werden soll: Siehe Abschnitt **stratificationField (optional, wird aber empfohlen)** auf Seite 71.

Bei Fertigstellung wird ein neuer Ergebnisordner mit einem Zeitstempel neben dem SAS-Datenset eingesetzt und die erstellten `report.html`- und `results.pdf`-Dateien werden automatisch geöffnet.

Diese Liste ausgewählter Variablen wird an SAS in den `&upliftSelVar`-Makro für die spätere Verwendung in SAS-Makroaufrufen zurückgegeben.

Beispielsweise wird folgendes Skript Variablen aus dem SAS-Datenset in `c:\data\DirectBank.sas7bdat` auswählen. Dabei wird die Variable `TrialRespond` als abhängige Variable und `TrialMailing` als Ansprechvariable verwendet, wobei die drei besten unabhängigen Variablen aus `Age`, `Income`, `Gender`, `MaritalStatus` und `CardVisa` ausgewählt werden:

```
libname testlib 'c:\data' ;

%uplift_variable_selection(
      data=testlib.DirectBank
     ,objectiveField=TrialRespond
     ,partitionField=TrialMailing
     ,analysisFields="Age,Income,Gender,MaritalStatus,CardVisa"
     ,nBestACs=3
)
```

Hinweis: Die Listen müssen durch Kommas getrennt und in Anführungszeichen aufgeführt werden.

Uplift-Modelle aus SAS heraus erstellen

Ein Makro `uplift_build_model` ermöglicht den Zugriff auf die Modellerstellungsfunktionalität innerhalb von Portrait Uplift. Dazu werden folgende Parameter benötigt:

data (Erforderlich: Dateipfad) Speicherort des SAS-Datensets (`.sas7bdat`) für die Analyse.

objectiveField (Erforderlich: Variablenname) Name der abhängigen Variable für die Analyse: Siehe Abschnitt **objectiveField (erforderlich)** auf Seite 70.

partitionField (Erforderlich: Variablenname) Name der Experimental-/Kontrollgruppenaufteilungsvariable: Siehe Abschnitt **partitionField (erforderlich)** auf Seite 71.

analysisFields (optional: eine durch Komma getrennte Liste der Variablennamen) Liste der für die Modellerstellung zu berücksichtigenden unabhängigen Variablen. Dies könnte das Makro `&upliftSelVar` sein, wenn es nach `%uplift_variable_selection` ausgeführt wird. Wenn dieser nicht vorhanden ist, werden alle Variablen berücksichtigt.

xanalysisFields (optional: eine durch Komma getrennte Liste der Variablennamen) Liste der aus der Berücksichtigung für die Modellerstellung auszuschließenden unabhängigen Variablen.

categoricalFields (optional: durch Komma getrennte Liste der Variablennamen) Liste der Variablen, die als nominale Variablen interpretiert werden sollen.

useExisting (optional: 0 oder 1) Markierung wird verwendet, um bestehende Dateninterpretationen und Klasseneinteilungen vom vorherigen Durchgang der Variablenauswahl zu benutzen.

metadata (optional: Dateiname) Portrait Miner-Metadatendatei (`.qsfm`), um die auf die Daten vor der Modellerstellung anzuwendenden Interpretationen zu definieren.

params (optional: Dateiname) Portrait Uplift-Parameterdatei (`.xml`) die bevorzugt auf die Makroparameter angewendet werden soll.

nBagsValues (optional: Integer-Liste in eckigen Klammern) Liste der Anzahl von Bags: Siehe Abschnitt **nBagsValues (erforderlich)** auf Seite 77.

pruneThresholdValues (optional: Prozentsatz-Liste in eckigen Klammern) Liste der zu verwendenden Pruning-Schwellenwerte: Siehe Abschnitt **pruneThresholdValues (erforderlich)** auf Seite 78.

autoPrune (optional: 0 oder 1): Siehe Abschnitt **autoPrune (optional; Standard: 0)** auf Seite 79.

nRuns (optional: Integer) Anzahl Durchläufe für jeden Parametersatz: Siehe Abschnitt **nRuns (optional; Standard: 1)** auf Seite 78.

nLevels (optional: Integer) Maximale Tiefe der erstellenden Bäume: Siehe Abschnitt **nLevels (optional; Standard 4)** auf Seite 74.

maxTreeAttempts (optional: Integer) Maximale Anzahl Versuche für jeden Baum: Siehe Abschnitt **maxTreeAttempts (optional; Standard: 2)** auf Seite 79.

minLeafNodes (optional: Integer): Siehe Abschnitt **minLeafNodes (optional; Standard: 4)** auf Seite 79.

autoMinPopSize (optional: 0 oder 1): Siehe Abschnitt **autoMinPopSize (optional; Standard 1)** auf Seite 73.

searchMinPopSize (optional: 0 oder 1): Siehe Abschnitt **searchMinPopSize (optional; Standard 0)** auf Seite 74.

minPopSize (optional: Integer): Siehe Abschnitt **minPopSize (optional; Standard 2000)** auf Seite 73.

propAnalysisCandidates (optional: Prozentsatz) Anteil der zu verwendenden unabhängigen Variablen: Siehe Abschnitt **propAnalysisCandidates (optional; Standard: 1.0)** auf Seite 79.

saveFocusStem (optional: String) Dateinamenstamm von gespeicherten Zwischendatensets: Siehe Abschnitt **saveFocusStem (optional; Standard: (leer))** auf Seite 82.

saveTreeStem (optional: String) Dateinamenstamm von gespeicherten Zwischenbäumen: Siehe Abschnitt **saveTreeStem (optional; Standard: (leer))** auf Seite 82.

tvTrainProp (optional: Prozentsatz) Anteil der für den Trainingssatz verwendeten Daten: Siehe Abschnitt **tvTrainProp (optional; Standard: 50%)** auf Seite 81.

modelType (optional: `auto` | `binary` | `real` | `twostage`): Siehe Abschnitt **modelType (erforderlich)** auf Seite 70.

yQiniAxis (optional: String) Text, der als Bezeichnung der vertikalen Qini-Achse verwendet werden soll: Siehe Abschnitt **yQiniAxis (optional, standardmäßiger kumulativer Uplift (pc pt))** auf Seite 73.

outcomeSizeActual (optional: Variablenname) Variable für das aktuelle Ergebnis: Siehe Abschnitt **outcomeSizeActual (optional)** auf Seite 70.

outcomeSizeEstimate (optional: Variablenname) Variable für das geschätzte Ergebnis: Siehe Abschnitt **outcomeSizeEstimate (für zweistufige Modelle erforderlich)** auf Seite 70.

championModelField (optional: Variablenname) Variable für das Champion-Modell zum Vergleich: Siehe Abschnitt **championModelField (optional)** auf Seite 72.

prngInitialState (optional: String) Der zu verwendende Startwert. Siehe Abschnitt **prngInitialState (optional)** auf Seite 81.

roiCurrencySymbol (optional: String): Siehe Abschnitt **roiCurrencySymbol (optional; Standard: $)** auf Seite 83.

roiCostOfTreatment (optional: String): Siehe Abschnitt **roiCostOfTreatment (optional; Standard: 0.00)** auf Seite 83.

roiValueOfBenefit (optional: String): Siehe Abschnitt **roiValueOfBenefit (optional; Standard: 1.00)** auf Seite 83.

roiCostOfFulfillment (optional: String): Siehe Abschnitt **roiCostOfFulfillment (optional; Standard: 0.00)** auf Seite 83.

Hinweis: Prozentsätze müssen das Prozentzeichen enthalten, z.B. `50%`.

Hinweis: `nBagsValues`- und `pruneThresholdValues`-Listen müssen in doppelte Anführungszeichen und rechteckige Klammern gesetzt werden, z.B. `"[1%,2%]"`.

Nach Abschluss wird neben dem SAS-Datenset ein neuer Ergebnisordner mit Zeitstempel angesiedelt und SAS wird drei Berichte mit Modellergebnistabellen und Qini-Diagrammen im PDF-, Excel- und HTML-Format erstellen. Eine Vorschau des HTML-Berichts wird im SAS Results Viewer angezeigt.

Zum Beispiel wird folgendes Skript Uplift-Modelle vom SAS-Datenset `c:\data\DirectBank.sas7bdat` erstellen. Dabei wird die Variable `TrialRespond` als abhängige Variable und `TrialMailing` als Ansprechvariable verwendet, wobei mithilfe der unabhängigen Variablen `Age`, `Income`, `Gender`, `MaritalStatus` und `CardVisa` insgesamt vier Modelle erstellt werden:

```
libname testlib 'c:\data' ;

%uplift_build_model(
      data=testlib.DirectBank
     ,objectiveField=TrialRespond
     ,partitionField=TrialMailing
     ,analysisFields="Age,Income,Gender,MaritalStatus,CardVisa"
     ,nBagsValues="[1,2]"
     ,pruneThresholdValues="[1%,2%]"
     ,nRuns=1
     ,nLevels=3
)
```

Uplift-Modelle in SAS bewerten

Ein Makro `uplift_score_model` bietet Zugriff auf die Modellauswertungs-Funktionalität innerhalb von Portrait Uplift. Dazu werden folgende Parameter benötigt:

data (Erforderlich: Dateipfad) Speicherort des zu bewertenden SAS-Datensets (`.sas7bdat`).

model (Erforderlich: Dateipfad) Speicherort des SAS-Modellcodes (`.sas`) für die Bewertung.

outputData (optional: Dateipfad) Speicherort des bewerteten SAS-Datensets (`.sas7bdat`). Falls das bewertete Datenset fehlt, wird es als eine Version der Quelldaten mit Zeitstempel erstellt.

outputFields (optional: Liste der Variablennamen) Liste der Variablen, die im bewerteten Datenset zusätzlich zu den Auswertungsfeldern erstellt werden.

force (optional: 0 oder 1) Erzwingt das Überschreiben eines vorhandenen Ausgabedatensets.

Nach Abschluss wird ein neues SAS-Datenset erstellt.

Zum Beispiel wird das folgende Skript das SAS-Datenset in `c:\data\DirectBank.sas7bdat` mithilfe des SAS-Uplift-Modellcodes in `c:\data\results20080208-165048\models\modelB02T1,00pcR0001.sas` bewerten und ein neues Datenset in `c:\data\results.sas7bdat` mit den Variablen `HomePostcode`, `Age` und `Gender` erstellen (ein bestehendes Datenset mit gleichem Namen wird dabei überschrieben).

```
libname testlib 'c:\data' ;

%uplift_score_model(
     data=testlib.DirectBank
   , model="C:\data\results20080208-165048\models\modelB02T1.00pcR0001.sas"
   , outputData=testlib.results
   , outputFields="HomePostcode, Age, Gender"
   , force=1
)
```

Kapitel 3

Ansatzüberblick

In diesem Abschnitt:

- Einleitung 30
- Modellstruktur 30
- Kampagnen-Design für Uplift-Analyse 31
- Datenvorbereitung 32
- Ansprechverzerrung und Modellvalidierung 32
- Mehrfachansprachen 35

Einleitung

Portrait Uplift implementiert einen Best-Practice-Ansatz für die Uplift-Modellerstellung. Der Ansatz selber enthält zwei breit angelegte Phasen: Variablenauswahl, welche die Erzeugung von Funktionen durch eine optimierte Klasseneinteilung beinhaltet, sowie einen Variablenausschluss (siehe Abschnitt **Ansatz für Variablenauswahl und Klasseneinteilung** auf Seite 37) und Modellerstellung (siehe Abschnitt **Modellerstellungsansatz** auf Seite 45). Ein Leitfaden zum besseren Verständnis der Modellergebnisse und der Bewertung der Modellqualität ist in Abschnitt **Modellbewertungsansatz** auf Seite 53 enthalten. Eine Übersicht der wichtigsten Parameter für das Modellieren befindet sich im Abschnitt **Parameterrichtlinien** auf Seite 61 und eine umfassende Referenz im Abschnitt **Parameter-Referenz** auf Seite 67.

Die Uplift-Modellierung ist für verschiedene Modellierungsszenarien, einschließlich binärer Ergebnisse, stetiger Ergebnisse und zweistufiger Modelle, wie im Abschnitt **Modellstruktur** auf Seite 30 beschrieben, geeignet. Bevor Sie Portrait Uplift anwenden müssen geeignete historische Kampagnendaten bereitgestellt werden. Idealerweise findet die Kampagnenplanung im Vorfeld statt, um die Effektivität späterer Uplift-Modellierungen zu maximieren. Empfehlungen für diese "Entwurf zur Analyse"-Herangehensweise werden im Abschnitt **Kampagnen-Design für Uplift-Analyse** auf Seite 31 vorgestellt. Nachdem ein Modellierungsdatenset zur Verfügung steht muss es für Portrait Uplift, wie im Abschnitt **Datenvorbereitung** auf Seite 32 beschrieben, vorbereitet werden. Falls keine konkrete Planung erfolgte, was häufig der Fall ist, ist es besonders wichtig zu überprüfen, ob die Modellierungsdaten die Annahmen von Portrait Uplift erfüllen. Insbesondere muss sichergestellt werden, dass es keine Verzerrung in der Experimental-/Kontrollgruppenaufteilung gibt, damit die Modellprognosen nicht ungültig werden. In einigen Fällen können Verzerrungen mit Hilfe einer Gewichtung, wie im Abschnitt **Ansprechverzerrung und Modellvalidierung** auf Seite 32 beschrieben, korrigiert werden.

Modellstruktur

Portrait Uplift unterstützt drei Modellstrukturen:

- Binäres Ergebnis ("`binary`");
- Stetiges oder geordnetes diskretes Ergebnis ("`real`");
- Zweistufig ("`twostage`").

Binäre Ergebnismodelle werden für Probleme verwendet, bei denen es genau zwei mögliche Ergebnisse gibt, z.B. gehen oder bleiben (Kundenbindung/Absprungquoten/Kundenabwanderung), kaufen oder nicht kaufen (vgl. "Resonanz"), oder Standard oder nicht Standard (Kreditrisiko). Dieses Ergebnis muss durch eine *1/0* Integer-Variable widergespiegelt werden. In diesen Fällen sagt das Uplift-Modell die Änderung der Wahrscheinlichkeit des 1-Ergebnisses für jeden Kunden vorher, z.B. Reduzierung der Abwanderungswahrscheinlichkeit, Anstieg der Resonanzwahrscheinlichkeit, Anstieg der Ausfallwahrscheinlichkeit usw.

Stetige ("reell") Modelle werden für Probleme verwendet, bei denen das Ziel die Stimulation des Grades oder der Größe der Kundenaktivität ist, z.B. der Anstieg der Ausgaben der Kunden. In diesem Fall fügt das Uplift-Modell die Änderung des Ergebnisses als ein Ergebnis der Ansprache ein.

Zweistufige-Modelle sind mit den stetigen Modellen verwandt, nur das hier das Ziel einer Kampagne die Erhöhung der Wahrscheinlichkeit (oder Frequenz) der Kundenaktivität (z.B. Kauffrequenz) oder die Erhöhung des Grades der Aktivität durch das Ankurbeln der Kauf- oder Verwendungsfrequenz ist. In diesem Fall wird das Uplift-Modell verwendet, um die Steigerung der Wahrscheinlichkeit des Ergebnisses zu schätzen, was jedoch anschließend mit einer (gelieferten) Schätzung der Größe des Ergebnisses (`outcomeSizeActual`), sofern es dazu kommt, kombiniert wird, wenn die Leistung jedes Modells in einen Bericht zusammengestellt wird.

Es ist nicht immer offensichtlich, ob die Hauptwirkung einer Anreizbehandlung zum Anstieg der Größe der Kundeneinkäufe oder der Wahrscheinlichkeit, dass der Kunde kauft, führen wird. Oft kann eine Einsicht gewonnen werden, indem sowohl der Lift der Kaufwahrscheinlichkeit als auch der Lift der Ausgabenrate für Kunden getrennt untersucht werden. Wird der allgemeine Lift der Ausgaben hauptsächlich durch einen Lift der Kaufrate erklärt (multipliziert mit einem durchschnittlichen Kaufbetrag), dann ist eher ein zweistufiges Modell angebracht. Ist der Lift der Kaufwahrscheinlichkeit annähernd Null, ist ein stetiges Modell die bessere Wahl. Unter anderem Umständen kann es ratsam sein sowohl ein "reelles" als auch ein zweistufiges Uplift-Modell auszuprobieren und die beiden möglicherweise zu kombinieren.

Kampagnen-Design für Uplift-Analyse

Die empfohlene Vorgehensweise für den Entwurf einer Kampagne, die durch das Modellieren des Uplifts analysiert werden soll, ist:

1. Kandidaten-Untersuchungsgesamtheit *P* zum Testen auswählen.
2. *P* zufällig in eine Experimental- und eine Kontrollgruppe unterteilen. Stratifizierung (exakte Größe Zufallsstichprobennahme) kann verwendet werden und hilft wahrscheinlich bei der Reduzierung von Lärm.
3. Das Modellieren ist am einfachsten, wenn die gesamte angesprochene Untersuchungsgesamtheit eine identische Ansprache erfährt oder wenn Ansprachen zufällig zugewiesen werden. Ist dies nicht der Fall, ist das Modellieren häufig dennoch möglich, wird aber komplexer und möglicherweise weniger genau sein. Siehe Abschnitt **Mehrfachansprachen** auf Seite 35.

Idealerweise wird die kleinere der beiden Gruppen (in der Regel die Kontrollgruppe) nach der folgenden Regel bemessen:

"Damit Uplift-Bäume zufriedenstellend erstellt werden und in der Lage sind, Uplift-Abweichungen von *x*% aufzuspüren, sollten wir sicherstellen, dass sowohl die Experimental- als auch die Kontrollgruppe groß genug sind, damit *x*% davon ungefähr 1,000 Personen repräsentieren. Falls wir also Modellabweichungen von 1% im Uplift abbilden möchten, benötigen wir Kontroll- und Experimentalgruppen von mindestens 100,000."

Dennoch ist sich Portrait der Tatsache bewusst, dass diese Regel häufig verletzt wird. Portrait Uplift wurde erarbeitet, um den größtmöglichen Wert aus Daten zu ziehen, selbst wenn diese Regel deutlich verletzt wird.

Datenvorbereitung

Vor der Uplift-Analyse muss das Modellierungsdatenset entsprechend vorbereitet werden:

- Markieren Sie das Feld der Zielvariable, entweder ein numerisches 0/1-Feld für ein binäres Modell (z.B. Resonanz, Abwanderung) oder ein reellwertiges Feld für ein stetiges Modell (z.B. Ausgaben).
- Markieren Sie das Zerlegungsfeld, ein numerisches 0/1-Feld, wobei 1 für die angesprochenen Kunden steht und 0 für Kunden der Kontrollgruppe. Für Optionen zur Analyse von Kampagnen mit mehreren Ansprachen, siehe Abschnitt **Mehrfachansprachen** auf Seite 35.
- Beachten Sie jedes Top-Modell-Feld, jedes numerische Feld, das Kunden nach Lift sortieren soll (höhere Werte, die auf einen höheren Lift hindeuten). Dabei kann es sich um ein alternatives Lift-Modell, ein traditionelles Modell oder eine Segmentierung, die zuvor zur genauen Zielgruppenansprache verwendet wurde, handeln.
- Fügen Sie bei Bedarf kategoriale Interpretationen zu den Analysekandidaten hinzu. Wir empfehlen kategoriale Werte als Integer-Daten anstelle von Strings zu speichern, da Letztere wesentlich höhere Speicheranforderungen haben.
- Führen Sie alle erforderlichen Datensatzauswahlen durch, um die Untersuchungsgesamtheit für das Modellieren zu identifizieren. NULL-Werte in der Zielvariable, Ansprache und Zerlegungsfeld sollten normalerweise ausgeschlossen sein. (NULL-Werte in Analysekandidaten werden korrekt analysiert und können sogar in eine separate Kategorie während des Baumwachstums aufgeteilt werden.)
- Fügen Sie den Analysekandidaten optional entsprechende Klasseneinteilungen hinzu, falls Sie keine automatische Optimierung der Klasseneinteilungen planen; siehe Abschnitt **Ansatz für Variablenauswahl und Klasseneinteilung** auf Seite 37.
- Erstellen oder beachten Sie optional eine festgelegte Trainings-/Validierungsaufteilung, die String-Werte "T" und "V" enthält. Siehe Abschnitt **fixedTrainValidateField (optional)** auf Seite 71.
- Erstellen oder beachten Sie optional ein Fallgewichtungsfeld, das nicht negative, numerische Werte enthält. Siehe Abschnitt **Ansprechverzerrung und Modellvalidierung** auf Seite 32.

Ansprechverzerrung und Modellvalidierung

Uplift-Modellierung setzt voraus, dass der *einzige* systematische Unterschied zwischen der Experimental- und der Kontrollgruppen-Gesamtmenge die zu modellierende Ansprache ist. Wenn die Ansprachenaufteilung nicht zufällig ist (wenn es beispielsweise möglich ist, den Ansprechindikator mit Hilfe eines konventionellen Modells vorherzusagen), sind die Ergebnisse des Modellierens eines Uplifts nicht verlässlich.

Portrait Uplift bietet die automatisierte Zerlegungsvalidierung (oder "Lila" Bericht, Abschnitt **Zerlegungsvalidierung (der "Lila" Bericht)** auf Seite 43), um zu testen, ob der Ansprechindikator bei jedem Analysekandidaten zufällig erscheint, um den Benutzer vor möglichen Problemen in diesem Bereich zu warnen.

Viele Arten bekannter Verzerrungen im Ansprechindikator können korrigiert werden, indem ein Gewichtungsfeld bestimmt wird, das den Wert einer Fallgewichtung für jeden Datensatz definiert (siehe Abschnitt **userWeightingField (optional)** auf Seite 72). Beispiele solcher Verzerrungen beinhalten:

Ansprechverzerrung

Oftmals werden unterschiedliche Teile des Kundenstamms in unterschiedlichen Intervallen angesprochen. Solange die Ansprache innerhalb eines Segments zufällig erfolgt, kann diese Verzerrung durch eine Fallgewichtung korrigiert werden. Stellen Sie sich beispielsweise vor, dass die Top-Drei-Dezile bei einer Rate von 90% angestrebt würden die unteren sieben Dezile bei einer Rate von 10%. Dies führt dazu, dass die Experimentalgruppendatensätze in den oberen Dezilen überrepräsentiert werden und in den unteren Dezilen unterrepräsentiert werden.

Im Allgemeinen wird angenommen, dass wir eine Ansprechrate haben, die (nur) entsprechend eines bekannten Segmentierungsfeldes mit einer allgemeinen Ansprechrate T und einer Ansprechrate T_s für jedes Segment s innerhalb dieses Feldes variiert. Wir können jeden Experimentalgruppendatensatz in s durch T / T_s und jeden (nicht angesprochenen) Kontrollgruppendatensatz in s durch $(1-T) / (1-T_s)$ gewichten. Nach der Fallgewichtung ergibt so eine konstante Ansprechrate in allen Segmenten und der allgemeine (gewichtete) Datensatzzähler bleibt in jedem Segment erhalten.

Sie können diese Art der Gewichtung als ein abgeleitetes Feld in Decision Studio unter der Annahme eines kategorialen Segmentierungsfeldes, namens `segment` und einem Ansprechindikator `treatment` einführen mithilfe von FDL, wie beispielsweise:

```
create weight := (
        q := mean(treatment);            // overall treatment rate
        p := mean(treatment) by segment; // rate for this segment
        if treatment then q/p else (1-q)/(1-p)
    );
```

Hinweis: In Fällen, wo die Ansprechverzerrung unbekannt ist und nur schwer in einem einzelnen Segmentierungsfeld angegeben werden kann, ist es mitunter dennoch möglich, die Verzerrung annäherungsweise durch Modellieren des Ansprechindikators zu korrigieren.

Bei diesem Ansatz wird ein traditionelles Regressionsmodell mit dem Ansprechindikator als Zielvariable erstellt (z.B. mit Hilfe des Scorecard Assistenten). (Dies kann häufig interessante Erkenntnisse über den Ursprung der Verzerrung ermöglichen.) Durch die Bewertung der Modellierungsdaten können wir eine "Wahrscheinlichkeit der Ansprache" für jeden Datensatz erstellen. Wir können diese Wahrscheinlichkeit anschließend direkt anstelle der Segment-Ansprechrate, T_s, wie oben beschrieben mit FDL verwenden, ähnlich:

```
create weight := (
        q := mean(treatment);           // overall treatment rate
        p := treatment_probability;     // as predicted by treatment model
        if treatment then q/p else (1-q)/(1-p)
    );
```

Es ist überflüssig zu erwähnen, dass in dieser Situation bei der Validierung und Interpretation von Modellierungsergebnissen besondere Vorsicht angebracht ist!

Zierverzerrung

Mitunter handelt es sich bei der für die Modellierung verfügbaren Untersuchungsgesamtheit um eine verzerrte Repräsentation der gesamten Untersuchungsgesamtheit, die später bewertet wird. In diesen Fällen ist es oft wünschenswert die modellierte Datei erneut zu gewichten, um ihre Genauigkeit bei der Bewertung zu erhöhen. Stellen Sie sich beispielsweise ein Dezil vor, das im gesamten Kundenstamm

bestimmt wurde (bei gleicher Kundenanzahl in jedem Dezil), wobei wir 90% der oberen drei Dezile (mit 10% als Kontrollgruppe markiert), und 9% der unteren sieben Dezile geplant haben (mit 1% als Kontrollgruppe). Dementsprechend haben wir eine konstante 90% Ansprechrate in der gesamten Datei und können sicher ein Uplift-Modell erstellen, wobei die Dezile der unteren Werte unterrepräsentiert sind und unser Modell an dieser Stelle dementsprechend ungenauer sein kann (in dem Sinne, dass unsere Modellqualität wesentlich stärker durch die Leistung in den oberen drei Dezilen beeinflusst wird).

Nochmals: wir gehen davon aus, dass die Zielverzerrung durch ein einzelnes Segmentierungsfeld mit einer Zielrate R_s in jedem Segment *s* (in unserem Beispiel ist *R=1* in den oberen drei Dezilen und *R=0,1* in den unteren sieben) erfasst wird. Wir gewichten einfach jeden Datensatz mithilfe von $1/R_s$, um die von der (verzerrten) Modellierungsdatei repräsentierte gesamte Untersuchungsgesamtheit widerzuspiegeln.

Für unser Beispiel könnten wir ein Fallgewichtungsfeld in Decision Studio mithilfe von FDL ableiten, wie dieser:

```
create weight := (
       case
         value_decile <= 3:     1.0;   // we have all high-value customers
         otherwise:             10.0;  // but only 10% of the lower deciles
    );
```

Kombination von Gewichtungen

Besondere Vorsicht ist geboten, wenn mehrere Gewichtungen (beispielsweise die Zielgewichtung und die Ansprechgewichtung) auf das gleiche Datenset angewendet werden müssen. Diese Gewichtungen sollten der Reihe nach und nicht unabhängig voneinander festgelegt werden.

Falls Sie ein Zielgewichtungsfeld *weight_target* haben, wird das abgeleitete Feld für die Ansprechverzerrung zu:

```
create t := treatment * weight_target; // weighted value for a treated record

  create weight_treat := weight_target * (
    q := sum(t) / sum(weight_target); // overall weighted treatment rate
    p := ( sum(t) by segment ) / sum(weight_target) by segment ;
      // treatment rate for this segment
    if treatment then q / p else (1-q) / (1-p)
  );
```

wieder unter der Annahme eines kategorialen Segmentierungsfeldes namens `segment` und Ansprechindikator `treatment`.

Stratifizierte Resonanzstichproben

Häufig wird es bei niedrigen Resonanzraten nicht für nötig gehalten alle Instanzen der Nicht-Reagierenden mit einzubeziehen. Zum Beispiel könnte die Datei für das Modellieren alle Reagierenden enthalten, aber nur 10% der Nicht-Reagierenden (Zufallsauswahl). Das zeigt keine Ansprech- oder Zielverzerrung, jedoch wird dadurch die Resonanzrate künstlich aufgeblasen. Es könnte eine Fallgewichtung eingesetzt werden um jeden Resonanzdatensatz als einzelnen Kunden und jede Nichtresonanz als 10 Kunden zu behandeln (1/10%). Das heißt,

```
weight := if response then 1 else 10;
```

alternativ kann die stratifizierte Datei ohne Fallgewichtung als Modell erstellt werden, wobei die endgültige Auswertung so angepasst wird, dass sie die Stratifizierung widerspiegelt. Falls ω den Anteil der tatsächlichen Nichtresonanzen in der Datei für das Modellieren (1/10 in unserem Beispiel) widerspiegelt,

dann kann die tatsächliche Uplift-Wahrscheinlichkeit p_{actual} auf die abgebildete Uplift-Wahrscheinlichkeitsbewertung *p* bezogen werden. Dies geschieht mithilfe von:

$$p_{actual} = \frac{\omega p}{1 - (1 - w)p}$$

Mehrfachansprachen

Portrait Uplift, so wie derzeit im Handel, wurde hauptsächlich dafür entwickelt, um den Fall einer einzelnen Aktion oder Ansprache einer zufällig ausgewählte Teilmenge einer potentiellen Zieluntersuchungsgesamtheit zu verarbeiten, wobei der verbleibende Rest dieser Untersuchungsgesamtheit der Kontrolle dient.

Natürlich möchten Unternehmen in der Realität eine Vielzahl von Ansprachen oder Aktionen gleichzeitig testen. Wir fassen diese Situation unter dem Oberbegriff *Mehrfachansprachen* zusammen.

Dabei werden verschiedene Fälle unterschieden.

- Der einfachste Fall ist der, bei dem es viele verschiedene Ansprachen gibt, die zufällig zugeordnet werden. Wobei das Ziel ist, zu verstehen, welche die beste Zieluntersuchungsgesamtheit und welche die richtige Ansprache ist und es nicht darauf ankommt, die Interaktion der beiden zu verstehen, das heißt, keine Ansprache auf Personen abzustimmen.

  Die einfachste Ansatz für diesen Fall ist beide Teile des Problems separat handzuhaben. Bündeln Sie alle Ansprachen und erstellen Sie ein Zerlegungsfeld mit 1 für Personen, die eine angesprochen wurden und 0 für jene, die nicht angesprochen wurden (die Kontrollgruppe). Erstellen Sie anschließend ein normales Uplift-Modell, das Ihnen sagt, welche Personen am positivsten durch eine Zufallsansprache beeinflusst wurden.

  Bewerten Sie unabhängig davon den gesamten Uplift im Vergleich zu den Kontrollgruppen, die mit den einzelnen Ansprachen verbunden sind. Dies erfolgt einfach durch die Berechnung des durchschnittlichen Ergebnisses `mean(objective)` für jede Ansprache und Subtraktion des durchschnittlichen Ergebnisses für die Kontrollgruppen.
- Der zweit einfachste Fall ist vielleicht der, bei dem die Zuordnung der Ansprachen zufällig ist, es aber das Ziel ist zu verstehen, welche Ansprache für welche Person die beste ist. Solange die Anzahl der Ansprachen klein bleibt, ist die bevorzugte Herangehensweise die, separate Datensets zu erstellen, die die einmal angesprochene Untersuchungsgesamtheit zusammen mit allen anderen Kontrollgruppen enthalten, sowie für jedes dieser Datensets ein separates Uplift-Modell zu erstellen. Jeder Kunde kann dann mit jedem Uplift-Modell bewertet werden und es können Ansprachen mit dem höchsten prognostizierten Uplift für diesen Kunden ausgewählt werden.

  Falls die Anzahl der Ansprachen höher ist und einige Ansprachen aber viel allgemeiner als andere sind, können die verbleibenden Ansprachen gruppiert werden.

  Diese Herangehensweise funktioniert nicht bei einem so genannten faktoriellen Design, bei dem es eine Reihe von zu ändernden Faktoren (z.B. Papiergröße, Papierfarbe und Rabattstufen) gibt, wenn die Anzahl der Kombinationen zu groß wird. Angesichts der n_i Auswahlmöglichkeiten für jede der *k*

Ansprechentscheidungen führt dies offensichtlich zu $\prod_{i=1}^{k} n_i$ Modellen. Falls *k* gleich 6 und jede Auswahl binär ist macht dies 2_6 = 64 Modelle erforderlich, was unrealistisch ist. Wir können jedoch von der vereinfachenden Annahme ausgehen, dass die Ansprechfaktoren voneinander unabhängig sind. Unter dieser Annahme können wir stattdessen ein unabhängiges Modell für jede Ansprachenauswahl erstellen und wir erhalten $\prod_{i=1}^{k} n_i$ Modelle statt sie zu erstellen. Mit 10 binären Faktoren wird die Modellanzahl von 64 auf 2 × 6 = 12 reduziert und weitere Faktoren der Reduzierung werden proportional größer. Die verschiedenen Modelle ermöglichen eine getrennte Auswahl für jeden Faktor von jedem Kunden ohne Berücksichtigung der Interaktionseffekte.

- Der komplexeste Fall ist der, bei dem es mehrere Ansprachen gibt, die nicht zufällig zugeordnet wurden.

  In diesem Fall ist es unmöglich zwischen dem Effekt der Ansprache und dem Effekt der genauen Zielgruppenansprache zu unterscheiden. Dennoch kann ein Uplift-Modell bei der Wahl, wer anzusprechen ist, hilfreich sein*, wenn die Strategie dazu verwendet wird um Ansprachen zuzuordnen*. Die Ansprechvariablen können als Analysekandidaten verwendet werden.

  Um das entstandene Modell zu verwenden berechnen Sie einfach, wie üblich, den vorhergesagten Uplift für jede Person und ordnen ihr anschließend ihre Ansprache auf der gleichen Grundlage zu, wie jene, die verwendet wird, um Ansprachen für die verwendete Kampagne zuzuordnen, um so das Uplift-Modell zu erstellen.

Kapitel

4

Ansatz für Variablenauswahl und Klasseneinteilung

In diesem Abschnitt:

- Einleitung 38
- Einfluss auf Modellstabilität 38
- Automatisierte Variablenauswahl 39
- Automatisierte Klasseneinteilung 40
- Ergebnisse und Berichte der Variablenauswahl 42

Einleitung

Portrait Uplift erstellt Uplift-Modelle mithilfe der im Modellierungsdatenset als Analysekandidaten markierten Felder. Die Klasseneinteilungen auf diesen Feldern bestimmen die Aufteilungspunkte, die während des Wachstumsprozesses des Baums berücksichtigt wurden. Daher können sowohl das Set der ausgewählten Analysekandidaten als auch die gewählten Klasseneinteilungen einen bedeutenden Effekt auf die Qualität der erstellten Modelle, wie im Abschnitt **Einfluss auf Modellstabilität** auf Seite 38 beschrieben, haben. Portrait Uplift kann sowohl den Prozess der Auswahl guter Kandidatenvariablen (Abschnitt **Automatisierte Variablenauswahl** auf Seite 39) als auch die Auswahl von Klasseneinteilungen "guter Qualität" für die Uplift-Prognose (Abschnitt **Automatisierte Klasseneinteilung** auf Seite 40) automatisieren. Während der Vorverarbeitungsphase werden eine Reihe diagnostischer Berichte erstellt, um Einsicht in die Datengültigkeit, Uplift-Stabilität und in feldweise Profile zu gewähren (Abschnitt **Ergebnisse und Berichte der Variablenauswahl** auf Seite 42).

Einfluss auf Modellstabilität

Sowohl die Variablenauswahl als auch die Klasseneinteilung helfen dabei, die Anzahl der Freiheitsgrade, die im nachfolgenden Modellerstellungsprozess erlaubt sind, zu steuern und in der Regel auch zu begrenzen. Das ist normalerweise sehr hilfreich, weil so einige Probleme im Zusammenhang mit der wiederholten statistischen Prüfung teilweise überwunden werden.

Die Variablenauswahl berücksichtigt nur Aufteilungen an Klassengrenzen für geordnete Variablen und behandelt die Kategorien der kategorialen Variablen als nicht teilbar, sodass die Auswahl der Klasseneinteilung von großer Bedeutung ist. Während es so aussehen mag, dass eine sehr feine Klasseneinteilung (mit vielen Klassen) der Software maximale Flexibilität bietet, als mit anderen Faktoren in Situationen mit großen Abweichungen. Häufig ist es jedoch besser eine Klasseneinteilung mit einer relativ geringen Anzahl von Klassen zu wählen, um die Wahrscheinlichkeit einer Überanpassung zu reduzieren.

Die nachfolgend beschriebenen Methoden zur Stabilitätsbewertung einer Variablen für die Uplift-Modellierung hängen von der verwendeten Klasseneinteilung ab, sodass die Variablenauswahl und die Auswahl der Klasseneinteilungen Hand in Hand gehen.

Als Faustregel für Situationen mit starken Abweichungen empfehlen wir mit einer groben Klasseneinteilung von vielleicht 5 Klassen gleicher Größe (20 %) zu beginnen. Wenn diese sich als stabil herausstellt kann es sich lohnen die Anzahl der Klassen zu erhöhen, vielleicht auf 10. Dabei muss darauf geachtet werden, dass die Untersuchungsgesamtheit nicht bis auf einen Punkt reduziert wird, an dem Uplift-Schätzungen unzuverlässig werden.

Selbst wenn eine optimierte Klasseneinteilung und Variablenauswahl unterstützt werden, wird dringend empfohlen, dass sich der Benutzer die Zeit nimmt die erstellten Berichte zu lesen und eine Variable nach der anderen daraufhin zu überprüfen, ob:

- die gewählte Klasseneinteilung stabil erscheint (benutzen Sie hierfür das Stabilitätsprofil);

- die Uplift-Abweichung bedeutsam, wesentlich und plausibel erscheint (benutzen Sie hierfür das Uplift Profil);
- die Zerlegung gültig ist (benutzen Sie hierfür die Zerlegungsvalidierung);
- Muster, die in den ausgewählten Feldern zu sehen sind und der angezeigte Uplift andere Variablen oder Kombinationen von Variablen, die für das Modellieren berücksichtigt werden sollten, empfiehlt.

Es wirkt zwar wie ein Klischee, von "garbage-in, garbage-out" zu sprechen, so bleibt dies doch eine interessante Beobachtung und nur der Benutzer kann die Ausgabe sowohl der Variablenauswahl als auch der Modellerstellung auf Plausibilität überprüfen. Ein positiverer Aspekt des Klischees ist, dass die Verbesserung der Qualität der Eingabedaten, die für das Modellieren verwendet werden, häufig messbare und signifikante Verbesserungen der Ausgabe-(Modell)-Qualität mit sich bringt.

Es ist unnötig zu erwähnen, dass während alle Aufmerksamkeit in diesem Dokument auf der Reduzierung der Anzahl der Freiheitsgrade liegt, die für das Modellieren zur Verfügung stehen, indem Analysekandidaten begrenzt und verhältnismäßig grobe Klasseneinteilungen verwendet werden, die Kehrseite dieser Vorgehensweise die ist, dass die potentielle Aussagekraft der Modelle reduziert wird. Da die Lärmsteuerung in den meisten Situationen bei der Uplift-Modellierung oberstes Thema zu sein scheint, ist es meistens viel nützlicher die Anzahl der Freiheitsgrade zu begrenzen, anstatt sie zu erweitern. Umgekehrt muss sich der Benutzer jedoch darüber im Klaren sein, dass es Fälle geben wird, in denen dies nicht der Fall ist und dass der Bogen überspannt werden kann.

Automatisierte Variablenauswahl

Portrait Miner benennt die unabhängigen Variablen, die für die Verwendung beim Modellieren *von Analysekandidaten zur Verfügung stehen.* In einigen Fällen ist es akzeptabel alle möglichen Analysekandidaten für die Baumerstellung zur Verfügung zu stellen und darauf zu vertrauen, dass die Bäume die passenden Kandidaten für die Einbeziehung in das Modell auswählen. In Situationen mit starken Abweichungen ist das jedoch meistens keine gute Idee, vor allem dann, wenn es eine besonders große Anzahl von Analysekandidaten gibt. Das Problem besteht darin, dass unter diesen Umständen häufig Aufteilungen gewählt werden, die nicht stabil genug sind. Pruning stellt sicher, dass instabile Aufteilungen aussortiert werden, wobei die [1] Die Natur der Methoden zur Baumerstellung bedeutet jedoch, dass die Möglichkeit einer Aufteilung an diesem Knoten dann verloren geht und das Ergebnis in der Regel ein schwaches Modell sein wird.

Portrait Uplift enthält daher eine Zusatzoption, die die *n* besten Analysekandidaten auf Grundlage einer konservativen Schätzung ihres prognostischen Werts, basierend auf der Qini-Metrik (siehe Abschnitt **Interpretation der Qini-Werte (Q und q0)** auf Seite 54), aus den verfügbaren Kandidaten auswählt.

Die Variablenauswahl verwendet die gleiche Struktur der XML-Parameterdatei wie die Modellerstellung und verfügt über einige zugeordnete Parameter (siehe Abschnitt **Parameter für die Variablenauswahl** auf Seite 74), insbesondere:

[1] so genannten "gierigen" Algorithmen versuchen werden Ihren Vorteil an jedem Entscheidungspunkt zu maximieren, was dazu führt, dass sie frühere Entscheidungen niemals erneut besuchen, bewerten oder ändern. Bei den meisten Methoden zur Baumerstellung gibt es eine gierige Erstellungsphase, während der sie auf jeder Stufe die beste Aufteilung zum Segmentieren eines Knoten wählen (entsprechend einiger Aufteilungskriterien). Eine spätere Pruning-Phase entfernt instabile Aufteilungen, ersetzt diese aber nicht durch Alternativen.

- `nBestACs`, die Anzahl an auszuwählenden Analysekandidaten;
- `saveBestACs`, der 1 ist, sobald die besten Analysekandidaten festgelegt wurden, sofern sie gespeichert werden sollen, und im anderen Fall 0 oder leer sind.

Automatisierte Klasseneinteilung

Portrait Uplift ist auch in der Lage Klasseneinteilungen für Felder auszuwählen. Genauso wie zu viele Analysekandidaten die Modellqualität beeinträchtigen können, können Bäume mit zu vielen Aufteilungen zu Variablen ähnlich abträglich sein. Dies gilt insbesondere in Situationen mit hohem Lärm, die dazu neigen die Uplift-Probleme der realen Welt zu bestimmen.

Der Parameter, der steuert, ob Felder erneut klassifiziert werden sollen, und falls ja, wie sie eingeteilt werden sollen, ist `setacsRebin`.

- Falls `setacsRebin` auf Null (0) gesetzt wurde, erfolgt keine neue Klasseneinteilung und es werden nur Analysekandidaten ausgewählt.
- Falls `setacsRebin` auf 1 (oder einen anderen nicht-Null-Integer) gesetzt wurde, wird die Software versuchen die Klassenanzahl für alle numerischen und kategorialen Analysekandidaten zu optimieren (außer Datumsfelder). Die Software wird außerdem eine XML-Datei namens `binnings.xml` in das Ergebnisverzeichnis (Abschnitt **Berichts- und Ausgabeparameter** auf Seite 81) schreiben, die später benutzt werden kann, um Klasseneinteilungen auf Feldern festzulegen.
- Letztendlich, falls `setacsRebin` auf den Namen einer XML-Klasseneinteilungsdatei gesetzt wurde, legt die Software die Klasseneinteilungen auf Felder fest, so wie in der Datei angegeben.

Optimierte Klasseneinteilungen

Die Software versucht Klasseneinteilungen für das Modellieren eines Uplifts zu optimieren, indem eine besondere Form des Low Qini Estimate für jede mögliche Klasseneinteilung jedes Feldes auswertet wird. Dabei werden konstante Klassengrößen mit (potentiell) 1 bis *M* (nicht Null) Klassen berücksichtigt, wobei *M* durch den Parameter `setacsRebinMaxBins` gesteuert wird, welcher standardmäßig 10 ist.

Im Prinzip ist die Arbeitsweise die gleiche wie bei einer Klasseneinteilung. Die (neu überarbeiteten) Daten werden zufällig auf acht Wege zerlegt, wobei die Klassen bei der ersten Zerlegung in absteigender Uplift-Reihenfolge sortiert werden. Danach wird bei dieser und jeder weiteren der sieben Zerlegungen der Qini (*Q*) ausgewertet. Für jede Klasseneinteilung werden dann vier Werte errechnet:

- *LQE* — der Low Qini Estimate ist $\mu(Q) - \sigma(Q)/\sqrt{n}$, wobei *n* die Anzahl an Beobachtungen ist (8 im Basisfall)
- $\mu(Q)$ — der durchschnittliche Qini-Wert
- MinSE — der kleinste Standardfehler ($\sigma / \sqrt{n}$).
- TPAQ — der "turning-point adjusted" Qini, wobei es sich um eine Variante der Qini-Berechnung handelt, die Umkehrpunkte in der Qini-Kurve bestraft.

Im Prinzip können wir dann eine Klasseneinteilung auf Grundlage von auf jedem dieser Werte erstellen. Das heißt mit dem höchsten LQE, dem maximalen Mittelwert-Qini, dem niedrigsten Standard-Fehler oder dem höchsten Turning-point adjusted Qini (TPAQ). Standardmäßig verwenden wir den höchsten

TPAQ, was aber mit Hilfe des Parameters `setacsRebinCriterion` (Abschnitt **setacsRebinCriterion (optional; Standard: T)** auf Seite 76) geändert werden kann.

Hierbei gibt es ein größeres Problem mit Verzerrung, das aber, wie wir herausgefunden haben, meistens mit Hilfe der Funktionen "Trimmen" und "Winsoring" behoben werden kann. Dies sind zwei Techniken zur Reduktion der Stichprobenabweichung. "Trimmen" beinhaltet das Verwerfen der *N* extremen Werte aus der Verteilung. "Winsoring" bedeutet, dass die gleichen extremen Werte bearbeitet und am Wert bei Erreichen (oder genau innerhalb) des Cutoff-Wertes "befestigt" werden. Da wir stets genau acht Beobachtungen haben, ist die Anwendung von Trimmen lediglich dazu gedacht den niedrigsten und höchsten Wert zu verwerfen, und die Anwendung von Winsoring dazu, den größten Wert an den des zweitgrößten zu heften, sowie den kleinsten an den zweitkleinsten. Ein weiterer Parameter, `setacsRebinMean`, steuert dieses Verhalten und kann auf `Normal`, `Trim` oder `Winsorize` gesetzt werden.

Wie man sehen kann, wird die Anzahl an Klassen für Klasseneinteilungen von 1 bis 10 Klassen zusammen mit der horizontalen Achse angezeigt und die Höhe der Punkte zeigt einen Qini-Wert. Jeder graue Kreis repräsentiert eine einzelne Beobachtung. (Wie dem Untertitel zu entnehmen ist, wurde hier das Trimmen angewendet, um die Anzahl der Beobachtungen von acht auf sechs zu reduzieren.) Der schwarze Punkt für jede Klasseneinteilung repräsentiert den Mittelwert *Q* und die Fehlerbalken zeigen einen Standardfehler (in diesem Fall $\sigma(Q)/\sqrt{6}$) an. Drei Merkmale werden rot hervorgehoben. Der Kreis am unteren Ende des Fehlerbalkens auf neun Klassen ist der höchste Low Qini Estimate. Die voll ausgefüllte rote Scheibe ist der höchste Qini-Mittelwert, der hier auch für 9 Klassen angezeigt wird. Der rote Fehlerbalken ist der kleinste Fehler, der natürlich aus der kleinsten Abweichung resultiert.

Die roten Dreiecke am oberen oder unteren Ende des Diagramms markieren einige Datenpunkte, die jenseits der *y*-Achse liegen. In diesem Fall sind diese zwar sichtbar, meistens ist dies aber nicht der Fall.

In diesem Diagramm wird offensichtlich, dass trotz Trimmen, die Abweichung groß ist und es nicht unerheblich ist zu entscheiden, welche Klasseneinteilung für diese Daten die beste ist. Dies gilt insbesondere bei der Berücksichtigung des offensichtlich ungewöhnlichen Verhaltens bei 8 Klassen, weshalb ein Fall für 5, 7 oder 9 Klassen erstellt werden sollte.

Wiederverwendung von Low Qini Estimates

Ein offensichtlicher Nebeneffekt der Optimierung einer Klasseneinteilung ist, dass Low Qini Estimates für jedes Feld erstellt werden. Es ist möglich die Wiederverwendung dieser Vorhersagen zu erzwingen, wenn Analysekandidaten ausgewählt werden. Dies ist die Standardeinstellung. Dieses wird vom Parameter `setacsUseRebinLQE` (1 oder 0) gesteuert.

Bitte beachten Sie, dass die Variablenauswahl im Allgemeinen Low Qini Estimates mehrmals auswertet und dadurch recht unterschiedliche Ergebnisse durch die zwei Methoden erzielt werden können. Im Allgemeinen führt die Einstellung von `setacsUseRebinLQE` auf Null (0) dazu, dass stabilere Klasseneinteilungen entstehen. Daher wird dies für besondere Lärmsituationen empfohlen.

Bitte beachten Sie auch, dass die optimierte Klasseneinteilung den Wert `setacsSigmaWeight` nicht berücksichtigt und die LQEs immer ähnlich $\mu - \sigma/\sqrt{n}$ sind.

Ergebnisse und Berichte der Variablenauswahl

Nach der Variablenauswahl kann der Befehl `qsupliftresults` im Ergebnisverzeichnis verwendet werden, um einen oder mehrere Berichte zu erstellen. Der Hauptbericht enthält eine Reihe selbsterklärender Abschnitte und einige, die der Erklärung bedürfen.

Es gibt in diesem Bericht drei Haupttabellen, nämlich:

- Ein Uplift Profil, welches zeigt, wie der Uplift mit jedem Feld variiert.
- Eine Zerlegungsvalidierung, die zeigt, wie stabil und somit gültig die Aufteilung der angesprochenen Kontrollgruppe ist.
- Ein Feldstabilitätsprofil, welches die Stabilitätsbewertung der Klasseneinteilung auf ein Feld im Kontext des Uplifts ermöglicht.

Die drei Berichte können auch aus einem Datenset mithilfe der Option `-p` und dem Befehl `qsupliftplus` erstellt werden. Die Form des Befehls ist

```
qsupliftplus -p params.xml
```

wobei der Focus, die Zielvariable und das Zerlegungsfeld und weitere Dinge in der Parameterdatei bestimmt werden `params.xml`, wie gewöhnlich (Abschnitt **Portrait Uplift von der Befehlszeile aus ausführen** auf Seite 18).

Das Ergebnis (einmal mit `qsupliftresults` bearbeitet) ist eine Datei `reportJJJJMMTT-HHMMSS.html`, wobei `JJJJMMTT` ein achtstelliges Datum und `HHMMSS` ein Zeitstempel ist, z.B. `report20121212-121212.html`. Diese Datei enthält die drei Berichte für den durch den Parameter `focusName` spezifizierten Focus.

Bei der Ausführung der Variablenauswahl kann mithilfe der Parameter `setacsProfile` gesteuert werden, ob diese Berichte für den Eingabefocus, den Ausgabefocus oder für beide erstellt werden sollen. Mit anderen Worten, ob alle Analysekandidaten oder nur Analysekandidaten abgedeckt werden sollen, die durch die Variablenauswahl für die Modellerstellung ausgewählt wurden). Die Werte für diesen Parameter sind `both`, `input`, `output` oder `none`. Er kann auch leer gelassen werden, was der Auswahl von `none` entspricht.

Uplift Profile

Uplift Profile zeigen den tatsächlichen Durchschnitts-Uplift als Funktion von jedem Feld an. Ein Beispiel hierzu finden Sie in der Abbildung **4**. Die Felder werden in der Reihenfolge aufgelistet, in der sie im Datenset erscheinen.

Das Uplift Profil erleichtert das Verständnis der groben Abweichung des Uplift innerhalb von jedem Feld. Beachten Sie, dass ungeachtet der (beabsichtigten) visuellen Ähnlichkeit, diese Tabelle sich entscheidend von einem Auswertungsprofil (Abschnitt **Auswertungsprofile** auf Seite 58) unterscheidet, da sie den *tatsächlichen* und nicht den vorhergesagten Uplift anzeigt.

Zerlegungsvalidierung (der "Lila" Bericht)

Eine Zerlegungsvalidierung wird in der Abbildung **5** dargestellt. In diesem Bericht werden alle Klassen, in denen sich der Experimentalgruppen-/Kontrollgruppenanteil vom gesamten angesprochenen Anteil wesentlich unterscheidet, in verschiedenen Lila-Stufen hervorgehoben. (Die Felder werden in absteigender Reihenfolge der Abweichung der Experimental-/Kontrollgruppenanteile aufgelistet.) Das Vorhandensein einer oder mehrerer dunkellila markierter Zellen kann darauf hinweisen, dass ein Problem oder eine Verzerrung der Experimental-/Kontrollgruppenaufteilung aufgetreten ist, wodurch die Modellierungsergebnisse ungültig wären. Eine derartige Verzerrung kann unter Umständen durch eine Fallgewichtung korrigiert werden. Siehe Abschnitt **Ansprechverzerrung und Modellvalidierung** auf Seite 32.

Stabilitätsprofil (der "Trommelstock"-Bericht)

In Abbildung **6** wird ein Stabilitätsprofil dargestellt. Das Stabilitätsprofil bietet eine schnelle und effiziente Möglichkeit, um problematische Klasseneinteilungen zu identifizieren, die möglicherweise von Hand angepasst werden müssen. Jeder farbige Streifen zeigt den Uplift für ein zufälliges Segment der Untersuchungsgesamtheit an. Der Zweck ist es, zu verstehen, wie stabil die Klasseneinteilung in Bezug auf den Uplift ist. Bei einer stabilen Klasseneinteilung ähneln sich die Farben der Streifen innerhalb einer Klasse (und haben einen ähnlichen Uplift). Wie Sie sehen können, ist dies hier bei vielen Klassen der Fall. Dennoch gibt es ein paar Ausnahmen. Falls innerhalb einer Klasse viele Abweichungen vorkommen, kann dies darauf hindeuten, dass die Klasseneinteilung nicht stabil ist und dass die Anzahl der Klassen reduziert werden sollte.

Die Streifenanzahl wird mit dem Parameter `nStabilityPartitions` gesteuert. Standardmäßig ist er auf 3 eingestellt und muss sich in der Bandbreite von 2 bis 10 bewegen.

Kapitel 5

Modellerstellungsansatz

In diesem Abschnitt:

- Einleitung 46
- Stichproben und Stratifizierung 46
- Bagging 47
- Pruning 48
- Automatisierte Parameteruntersuchung 48
- Automatisierte Parameterempfehlung 49
- Endgültige Neuerstellung 50
- Modellverunreinigung 50

Einleitung

Nachdem Variablen ausgewählt und angemessen in Klassen eingeteilt wurden kann die Modellerstellungsphase beginnen. Diese beinhaltet zunächst eine Trainings- und Validierungszerlegung (falls nicht vom Benutzer festgelegt) mit sorgfältiger Stratifizierung, um Verzerrung zu minimieren (siehe Abschnitt **Stichproben und Stratifizierung** auf Seite 46). Innerhalb der Trainingsdaten wird erneut eine Stichprobennahme benutzt, um Datensets mit mehreren Varianten mit einem anderen Baummodell auf jeder Version zu erstellen und kombiniert, um ein Bagging-Modell zu erstellen (siehe Abschnitt **Bagging** auf Seite 47). Jedes Baummodell wird durch Überprüfung seiner Abweichung quer durch die Datensets der neuen Stichprobe gesäubert, wieder um Lärm zu reduzieren (siehe Abschnitt **Pruning** auf Seite 48). Die für das Wachstum der Bäume verwendeten Schlüsselparameter können sowohl manuell als auch automatisch untersucht werden, um die Modellerstellungsleistung anzupassen (siehe Abschnitt **Automatisierte Parameteruntersuchung** auf Seite 48), wobei die besten Parameter automatisch empfohlen werden (siehe Abschnitt **Automatisierte Parameterempfehlung** auf Seite 49). Häufig wird das Endmodell über das vollständige Modellierungsdatenset mithilfe der optimalen Parameter (siehe Abschnitt **Endgültige Neuerstellung** auf Seite 50) wiederhergestellt (ohne Validierungsdaten). Obwohl ein drastischer Trainings- und Validierungsprozess stattfindet, ist es immer wichtig potentielle Quellen der Modellverunreinigung zu berücksichtigen (siehe Abschnitt **Modellverunreinigung** auf Seite 50).

Stichproben und Stratifizierung

In Situationen mit starken Abweichungen ist es wichtig, dass wenn Daten für das Training verfügbaren sind die für Test-/Trainings-/Validierungszwecken zerlegt werden müssen, die aktuell für die Modellerstellung verwendet werden, im Ganzen repräsentativ sind. Aus diesem Grund bietet Portrait Uplift die Möglichkeit, (stratifizierte) Stichproben zu nehmen, um sicherzustellen, dass die Daten so repräsentativ wie möglich sind.

Wir verwenden die Terminologie:

- *Validierungsdaten.* Eine zufällige Teilmenge aller verfügbaren Daten, die während der Modellerstellung nicht verwendet wird. Die Validierungsdaten werden nur benutzt, um die Leistung dieser ("ausreichenden") Daten darzustellen.
- *Trainingsdaten.* Alle Daten, die nicht zu Validierungszwecken verwendet werden. Diese Trainingsdaten werden dann weiter in neue Stichproben-Trainings- und Testzerlegungen unterteilt. (Es gibt möglicherweise eine Verwechslung zwischen Trainingsdaten — also die gesamten Daten, die für Trainingszwecke zur Verfügung stehen — und neue Stichproben-Trainingsdaten — die Teilmenge der Trainingsdaten, die in der Wachstumsphase eines Baums verwendet wird.)
- *Testdaten.* Neue Teilmengen der Stichprobe der Trainingsdaten, die zur Bewertung der Baumleistung für Pruning verwendet wird.

Diese wird in Abbildung **7** dargestellt.

Stratifizierte Stichproben basieren auf einer *Stratifizierungsvariable*, welche das Modellierungsdatenset in Segmente zerlegt. Wenn die Trainings- und Validierungsaufteilung erzeugt wird, muss jede ausgeglichen

sein (über zufällige Stichprobennahme unter Beachtung der genauen Größe). Die Stratifizierungsvariable kann entweder automatisch als Teil des Prozesses Variablenauswahl erstellt werden oder durch den Benutzer bereitgestellt werden (siehe Abschnitt **stratificationField (optional, wird aber empfohlen)** auf Seite 71). In jedem Fall wird das Feld normalerweise mit einer Variablenableitung erstellt, wie

```
segindex() by Objective, Partition, BestPredictor
```

wobei Zielvariable und Zerlegung die entsprechend Zielvariablen- und Zerlegungsfelder bilden und BestPredictor ein Feld ist, das in hohem Maße vorhersagend für Lift ist. Falls der Top-Prädiktor wenige Klassen aufweist, können Sie auch über den nächsten Prädiktor stratifizieren, um so in der Stratifizierungsvariable auf eine Gesamtanzahl von (nicht leeren) Klassen zwischen 20 und 60 zu kommen.

Hinweis: Die Stratifizierung wird durch die Klasseneinteilung auf der Stratifizierungsvariable definiert und nicht durch die Werte. In der Praxis bedeutet dies, dass sie normalerweise kategorial gemacht werden sollte.

Hinweis: Wenn Sie eine Fallgewichtung verwenden (siehe Abschnitt **userWeightingField (optional)** auf Seite 72), können zufällige Stichproben unter Beachtung der genauen Größe nicht mit Sicherheit eine vollkommen ausgewogene Stichprobe ergeben, insbesondere bei starken Abweichungen der Fallgewichtungswerte. Da die Stichprobennahmen "von oben nach unten" vorgehen, kann es in diesen Fällen nützlich sein, das Modellierungsdatenset durch umgekehrte Gewichtung auszusortieren, um sicherzustellen, dass die größten gewichteten Datensätze zuerst erscheinen.

Bagging

Die von Portrait Uplift erstellten Modelle sind "bagged", das heißt, sie bestehen aus einer Kombination verschiedener individueller Baummodelle. Bagging ist eine Methode um zu versuchen mehr Daten für Modellierungszwecke erstellen zu lassen, wenn es knapp wird. Der Grundgedanke besteht darin, die *Neue Stichprobennahme* der Trainingsdaten *k* Mal zu durchzuführen und *k* Modelle zu erstellen, um dann ihre Ergebnisse irgendwie miteinander zu kombinieren — normalerweise durch einfache Durchschnittsberechnung. Dies führt häufig zu größerer Modellstabilität.

Bei einem Trainingsdatenset T der Größe N wird ein Strichproben-Datenset T^{train} der gleichen Größe erstellt, indem wiederholte Male ein zufällig ausgewähltes Mitglied der Trainingsdaten T gezogen wird, um es in der neuen Stichproben-Untersuchungsgesamtheit T^{train} abzulegen und anschließend in T zu ersetzen (Stichprobennahme mit Ersetzen). Dieser Vorgang wird N Mal wiederholt. Ohne Fragen werden einige Datensätze sich in T^{train} wiederholen und andere werden nicht vorhanden sein. Unterschiedliche Versionen neuer Stichproben des gleichen Datensets werden unterschiedliche Frequenzen für die Datensätze haben und somit die auf den verschiedenen neuen Stichprobenversionen erstellten Modelle dazu anhalten, sich "darauf zu konzentrieren", verschiedene Teile des Datensets anzupassen. Portrait Uplift führt eine Stichprobennahme mit Ersetzen durch, indem zufällige Fallgewichtungsfelder erstellt werden.

Portrait Uplift verwendet eine Variable `nBags`, um die Anzahl der zu kombinierenden Modelle zu bestimmen. Diese Variable wird durch den Benutzerparameter `nBagsValues` (Abschnitt **nBagsValues**

(erforderlich) auf Seite 77) festgelegt. Derzeit wird die endgültige Vorhersage durch die Kombination der individuellen Baumvorhersagen mit dem simplen arithmetischen Mittelwert erstellt.

Pruning

Die durch Portrait Miner bereitgestellte Pruning-Methode ist eine Variante der standardmäßigen Kosten-Komplexitätsbereinigung mithilfe von Qini als (umgekehrte) Kostenmaßnahme. Dies ist in Situationen starker Abweichungen, für die Portrait Uplift entwickelt wurde, nicht effektiv.

Portrait Uplift führt eine abweichungsbasierte Pruning-Methode durch, bei der Aufteilungen, die für untergeordnete Knoten sorgen und eine starke Uplift-Abweichung auf einen Satz neuer Stichprobenkopien der Trainingsdaten ausüben, bereinigt werden.

Pruning wird durch eine Reihe von Parametern gesteuert, die in Abschnitt **Pruning-Kontrollgruppenparameter** auf Seite 78 beschrieben werden. Der wichtigste Parameter ist die Variable `pruneThreshold`, die durch den Benutzerparameter `pruneThresholdValues` festgelegt wird (Abschnitt **pruneThresholdValues (erforderlich)** auf Seite 78).

Falls die Pruning-Schwellenwerte zu niedrig eingestellt sind werden übermäßig kleine Bäume erstellt. Die Software verfügt über weitere Parameter, die in dieser Situation eine Reihe von Neuerstellungen erlauben und den Pruning-Schwellenwert zunächst konstant halten, um ihn dann zu erhöhen.

Ein Parameter `autoPrune` (Abschnitt**autoPrune (optional; Standard: 0)** auf Seite 79) kann automatisch zum Festlegen eines einzelnen Wertes in `pruneThresholdValues` verwendet werden.

Automatisierte Parameteruntersuchung

Die Hauptschlaufe des erstellten Uplift-Modells hat folgende Form (in Pseudo-Code):

```
for nBags in nBagsValues:
    for minPopSize in minPopSizeValues
        for pruneThreshold in pruneThresholdValues:
            for i = 1, 2, ..., nRuns:
                Build Uplift Model
```

wobei `nBagsValues` ein Satz von Werten für `nBags` ist, `minPopSizeValues` ein intern erstellter Satz von Werten für `minPopSize` ist, `pruneThresholdValues` ein Satz von Werten für `pruneThreshold` ist und `nRuns` die Anzahl von Durchläufen ist und für jeden Wert der Anzahl von Bags und Pruning-Schwellenwerte ausgeführt wird.

Daher ist es einfach einen Satz von Durchläufen im Batch festzulegen und eine Parametersuche für Mindestgrößen der Untersuchungsgesamtheit, Pruning-Schwellenwerte und Anzahl an Bags auszuführen.

Automatisierte Parameterempfehlung

Portrait Uplift beinhaltet, sofern vorhanden, in der Datei `out.xml` eine Empfehlung der "besten" Modellparameter. Die empfohlenen Modellparameter werden wie folgt ausgewählt:

- Gibt es keine Validierungsdaten, werden keine Modellparameter empfohlen. Andernfalls wird jede Gruppe von erstellten Modellen, die die gleichen Parameter verwenden, analysiert für die Bestimmung von:
  - Wie viele der Modelle in der Untergruppe den *Validierungstest* bestehen. Ob ein Modell standardmäßig "gültig" ist, wenn der Validierungs-Qini innerhalb ±20% seines Trainings-Qini-Werts liegt.
  - Ob die Modelle in der Untergruppe einen "Konsistenztest" bestehen. Ob standardmäßig eine Gruppe von Modellen konsistent ist, wenn der Standardfehler der Qini-Validierungswerte weniger als 20% des Mittelwertes dieser Werte ist.
- Wenn mindestens n von m Modellen (wobei n 3 und m standardmäßig 4 ist) den Validierungstest bestehen und die gültigen Modelle kollektiv den Konsistenztest bestehen, ist die Gruppe ein Kandidat für "beste".

  Wenn keine Gruppe die genannten Kriterien erfüllt werden keine Modellparameter empfohlen.
- Wenn eine oder mehrere Gruppen die genannten Kriterien erfüllen wird die Gruppe mit den besten 'Qualitätswerten' für die Modelle, die den Validierungstest bestanden haben, als empfohlen markiert.

  Wenn der Benutzer, wie zuvor in diesem Dokument beschrieben, Finanzparameter bereitstellt, wird als "Qualitätswert" zur Beurteilung des besten Modells der höchste, mithilfe der Finanzparameter errechnete Profitwert (gemittelt über die Modelle, die den Validierungstest bestanden haben) verwendet. Diese Berechnung beinhaltet nicht den Endpunkt auf der Profitkurve (d.h. Ansprechen aller Kunden), da diese Zahl nicht bei der Unterscheidung zwischen den Modellen benutzt wird.

  Wenn der Benutzer keine Finanzzahlen bereitgestellt, wird der höchste durchschnittliche Validierungs-Qini verwendet, um das beste Modell anzuzeigen.

Wenn eine Empfehlung ausgesprochen wird, wird eine `recommended_params.xml`-Datei geschrieben, die für die Ausführung der endgültigen Modellerstellung mithilfe der "besten" identifizierten Parameter geeignet ist. Diese Datei wird zusammen mit 'out.xml' in das Ausgabeverzeichnis geschrieben und wird für den "endgültigen Neubau"-Schritt (siehe Abschnitt **Endgültige Neuerstellung** auf Seite 50) festgelegt, wie:

- nRuns = 1
- nBagsValues = [20]
- tvTrainProp = 1.0
- autoPrune = 0
- pruneThresholdValues = *der Schwellenwert aus dem besten Durchlauf*
- autoMinPopSize = 0
- searchMinPopSize = 0
- minPopSize = *die Größe* minPopSize *aus dem besten Durchlauf*

Endgültige Neuerstellung

Unser Ziel beim Modellieren ist es, die uns zur Verfügung stehenden Daten mit maximalem Effekt zu verwenden, um so das bestmögliche Modell zu erstellen. In der Regel entscheiden wir uns dafür, einige der Daten, die wir während der Modellerstellung zur "Validierung" verwenden könnten, zu opfern. Wir tun das, weil wir die Gefahr, die Daten, die wir unmittelbar bei der Modellerstellung benutzen, übermäßig anzupassen, kennen und vermeiden möchten. Bei dem Versuch Modelle zu erstellen, in denen die Leistung (insbesondere die Qini-Leistung) ähnlich der in unseren Trainings- und Validierungsdatensets ist, reduzieren wir das Risiko der Überanpassung auf Kosten dessen, dass wir dem Modellerstellungsprozess den Zugriff auf einige seltene Daten verweigern. Trotzdem andere Dinge gleich bleiben, erwarten wir nicht nur, dass wir in der Lage sind präzisere Modelle zu erstellen, sondern auch die Wahrscheinlichkeit der Überanpassung zu reduzieren, indem wir mehr Daten verwenden.

Wir empfehlen daher eine Trainings-/Validierungsaufteilung für einen Satz von Durchläufen während der Erkundung der Modellierungs-Hauptparameter zu verwenden und dabei den Pruning-Schwellenwert und die Mindestgröße der Untersuchungsgesamtheit für die Bäume einzuschließen. Wenn ein guter Parametersatz gefunden wurde, der zu einer konstant guten Validierungsleistung führt, empfehlen wir einen oder mehr endgültige Durchläufe, häufig mit einer größeren Anzahl an Bags (Bäume pro Modell), wobei das Trainingsverhältnis auf 100% festgelegt ist, sodass keine Validierungsdaten vorhanden sind. Hierfür können alle Parameter, außer die Mindestgröße der Untersuchungsgesamtheit, die in gleichem Maße wie die Trainingsdaten heraufgesetzt werden sollte, gleich bleiben. Zum Beispiel, wenn Sie das standardmäßige Trainingsverhältnis von 66% verwendet haben, dann sollten Sie die Mindestgröße der Untersuchungsgesamtheit mit ca. 1.5 (100/66) multiplizieren.

Der Zweck der Erstellung von mehr als einem "endgültigen" Modell ist zu überprüfen, ob die Leistung konstant bleibt. Im Idealfall sollten alle endgültigen Modelle ähnliche Auswertungen für die große Anzahl von Kunden ergeben.

Modellverunreinigung

Portrait Uplift wendet, wie im Abschnitt **Stichproben und Stratifizierung** auf Seite 46 beschrieben, einen rigorose Trainings-/Validierungs-Ansatz an. Dennoch muss die Modell-Kontaminierung noch besprochen werden. Diese tritt ein, wenn auf Daten von einem Validierungsset während der Modellerstellung zugegriffen wird, und kann die Widerstandsfähigkeit der Modelle verringern (in Bezug auf die Fähigkeit sich auf die Zukunft zu beziehen, ungesehene Daten). Oftmals ist die Kontaminierung sehr dezent und hat nur minimale Folgen. Dennoch sollte sich der Benutzer dieser Möglichkeit bewusst sein und entsprechende Maßnahmen ergreifen, um dem gegebenenfalls entgegenwirken zu können.

Hier einige mögliche Gründe für eine Kontaminierung:

- Wird ein Modell auf der Grundlage seiner Leistung bei den Validierungsdaten (Standardansatz) aus den Kandidatenmodellen ausgewählt, entsteht eine gewisse Form der Verunreinigung, da die Wahl des Modells durch die Validierungsdaten getroffen wurde. Dieser Effekt ist von geringer Bedeutung, solange nur eine geringe Anzahl von Modellen auf diese Weise verglichen wird. Theoretisch ist der bessere Ansatz Validierungsdaten über einen Satz vergleichbarer Durchläufe zu verwenden, um

Parameter für eine Modellerstellung auszuwählen und anschließend ein neues Modell über die gesamte Untersuchungsgesamtheit mithilfe dieser Parameter (wenn angemessen maßstabsgetreu) zu erstellen.

- Wir empfehlen eine Stratifizierung nicht nur der Ansprech- und Ergebnisvariablen, sondern auch mindestens einer Schlüssel-Treiber-Variablen (siehe Abschnitt **Stichproben und Stratifizierung** auf Seite 46). Ein natürlicher Weg, diese Treiber-Variable auszuwählen, ist die Daten zu überprüfen. Da das Ziel dieser Untersuchung die Bestimmung der Stratifizierungsvariablen ist, die bei der Definition der Trainings-/Validierungsaufteilung verwendet werden, gibt es hier ein unvermeidbares Element der Verunreinigung, obwohl es nur in sehr geringem Maße erwartet wird. Dennoch sollte dieser Vorgang theoretisch vermieden werden, falls jede Art der Verunreinigung abgelehnt wird. In diesem Fall müssten die Treiber-Variablen für die Stratifizierung auf einer anderen Grundlage ausgewählt oder verworfen werden.
- Technisch gesehen sollten Analysekandidaten auf der Grundlage der Leistung nur in Bezug auf die Trainingsdaten ausgewählt werden. Falls alle Daten (Training und Validierung) für die Variablenauswahl verwendet werden, entsteht eine gewisse Verunreinigung, die in der Regel aber nur sehr gering ist. Die Vorgehensweise der Stratifizierung oben auf dem Prädiktor macht die Vermeidung dieser minimalen Verunreinigung schwierig. Falls sich der Benutzer darüber Sorgen macht, sollte er Trainings- und Validierungsdaten in separate Dateien unterteilen und für die Variablenauswahl nur die Trainingsdatei verwenden. Eine kombinierte Datei mit einem Trainings-/Validierungsfeld kann dann zusammen mit dem Parameter `fixedTrainValidateField` benutzt werden, um der Software zu erlauben sowohl über die Trainings- als auch über die Validierungsleistung zu berichten.

Kapitel 6

Modellbewertungsansatz

In diesem Abschnitt:

- Einleitung 54
- Interpretation der Qini-Werte (Q und q0) 54
- Interpretation von Qini-Kurven 55
- Ergebnisse und Berichte 57

Einleitung

Die Hauptausgaben der Portrait Uplift-Modellerstellung sind

- Die Qini-Statistiken (für Training, Validierung und kombinierte Daten und optional auch für ein "Top"-Modell);
- Die Qini-Kurve (ein Uplift-Gains-Diagramm);
- Ein Modellbericht einschließlich einer grafischen Abbildung des Modells;
- Die bewerteten Kunden;
- FDL, das jedes Bagging-Modell widerspiegelt, gespeichert in der Portrait XML-Einbettung von FDL, so wie vom `qsderive`-Bewertungstool verwendet.

Qini-Diagramme und -Werte werden detailliert in einem separaten Dokument besprochen. Hier wird jedoch die Interpretation der Werte Q und q_0 besprochen.

Interpretation der Qini-Werte (Q und q_0)

Die Qini-Koeffizienten Q und q_0 sind Kennzahlen darüber, wie gut das Modell Kunden durch Uplift *sortiert*. Da wir die aktuelle Änderung des Ergebnisses für jeden individuellen Kunden nicht kennen, kann dies nur auf segmentierter Basis bewertet werden.

Der zutreffende Qini-Wert Q verläuft von 100% (für ein perfektes Modell) durch Null für ein Zufallsmodell und bis -100% für ein umgekehrtes perfektes Modell. Wegen der Wahrscheinlichkeit eines negativen Uplift in einigen Segmenten kann das theoretisch optimale Modell in mehr Resonanzen durch die Zielgruppenansprache eines Teils der Untersuchungsgesamtheit auftreten, als bei Ansprache der gesamten Untersuchungsgesamtheit. Aktuelle Modelle werden bewertet, indem der von der Qini-Kurve und der Diagonalen eingeschlossene Bereich durch den vergleichbaren Bereich für das optimale Modell dividiert wird. Daraus ergibt sich, dass zutreffende Qini-Werte für gute, realitätsnahe Uplift-Modelle oft sehr klein sind. Obwohl negative Effekte in der realen Welt auftreten können und das auch tun, treten sie im Allgemeinen sehr viel seltener auf als theoretisch möglich.

Portrait Uplift zeigt in vielen Fällen zwei Qini-Werte an — Q und q_0 . Der Wert q_0 wird genauso berechnet wie der Wert Q, mit Ausnahme, dass anstelle der Division durch den Bereich unter dem theoretischen Optimum der Wert q_0 durch die perfekte Uplift-Kurve dividiert wird *, vorausgesetzt, dass es keine negativen Effekte gibt*. (Aus diesem Grund nennen wir q_0 manchmal den "Zero Downlift Qini", weswegen wir die Indizierung 0 verwenden.)

Werte von q_0 sind mehr wie traditionelle Gini (ROC)-Werte, aber mit einem großen Unterschied: wenn es negative Effekte auf Teile der Untersuchungsgesamtheit *gibt*, können q_0 Werte 100% überschreiten (und unter -100% fallen).

Wenn der Gesamteffekt einer Ansprache über die gesamte Untersuchungsgesamtheit sehr klein ist (nahe Null), wird der Wert q_0 sehr groß, und wenn der allgemeine Uplift genau Null ist, wird q_0 normalerweise unendlich. In diesem Fall wird der ausgegebene Wert q_0 Null und bedeutungslos sein. Das ist einer der Gründe, warum das gesamte Qini-Ausmaß Q ebenfalls angegeben wird.

Seien Sie sich bewusst, da der Wert q_0 tatsächlich den allgemeinen Uplift als Teil seines Nenners hat, wodurch selbst kleine Änderungen beim Gesamt-Uplifts eine heftige Auswirkung auf Qini-Größen haben können und sie nicht einfach zwischen Problemen verglichen werden können.

Interpretation von Qini-Kurven

Die Qini-Kurve wird detailliert in einem Begleitdokument mit dem Titel "Der Qini-Maßstab der Leistung für Uplift-Modelle" beschrieben. Hier finden Sie auch einen Überblick über einige Schlüsselmerkmale.

Eine Gains-Kurve für Uplift

Grundsätzlich handelt es sich bei der Qini-Kurve um ein Gains-Diagramm für Uplift. Wie bei einem Standard-Gains-Diagramm wird die Untersuchungsgesamtheit zunächst in der Reihenfolge "bester" bis "schlechtester", basierend auf einem Modell, sortiert. Die horizontale Achse zeigt den Anteil der Zieluntersuchungsgesamtheit unter der Annahme an, dass die "besten" Personen als erstes anvisiert werden (also auf der linken Seite).

Die vertikale Achse zeigt die Schätzung des (normalisierten) kumulierten Gesamt-Uplifts an, der durch das Anvisieren einer Untersuchungsgesamtheit erreicht wird. Es ist entscheidend zu verstehen, dass dieser Uplift immer unter Berücksichtigung der ganzen Untersuchungsgesamtheit quantifiziert wird.

Vertikale Achse für Binäre Ergebnisse (pc pt)

Bei einem binären Modell wird der kumulierte Uplift als Prozentsatz der gesamten Untersuchungsgesamtheit ausgedrückt — also in "Prozentpunkten" des Uplifts. Zum Beispiel wird liegt ein Problem mit einer Untersuchungsgesamtheit von 100.000 vor, bei dem eine Resonanz ein Kauf ist und bei dem es eine allgemeine Kaufrate von 5% in der Kontrollgruppe gibt. Dann bedeutet ein Punkt bei *(30%, 10%)* auf einer Qini-Kurve, dass wenn die besten 30% der Untersuchungsgesamtheit anvisiert werden, ein Uplift von 10 Prozentpunkten auf die *gesamte* Untersuchungsgesamtheit erreicht wird, also eine allgemeine Kaufrate von *5% + 10% = 15%* von 100.000 = 15,000, ein Anstieg von 10.000 Käufen.

Damit ist *kein* Uplift von 10% für 30% der Untersuchungsgesamtheit gemeint, sondern nur 3,000 zusätzliche Käufe.

Vertikale Achse für stetige Ergebnisse (photp)

Ähnlich bei einem stetigen Ergebnis, z.B. die Kaufgröße, wird auf der vertikalen Achse der durchschnittliche Uplift pro Kopf der *Gesamten* Untersuchungsgesamtheit angezeigt. Ein Punkt auf der Qini-Kurve bei (30%, $35) bedeutet, dass wenn die besten 30% der Untersuchungsgesamtheit anvisiert würden, es einen Uplift von $35 pro Kopf der Untersuchungsgesamtheit geben würde.

Bei einer Untersuchungsgesamtheit von 100.000 und einer durchschnittlichen Kaufgröße von $100 in der Kontrollgruppe beträgt der Gesamtumsatz ohne anvisierte Zielgruppe $100 × 100.000 = $10.000.000. Würden 30% anvisiert, würde der durchschnittliche Kaufwert auf $100 + $35 = $135 ansteigen, sodass sich der Gesamtumsatz auf $135 × 100 000 = $13 500 000 und die Steigerung auf $3,5 Millionen belaufen würde.

Nochmal, damit ist kein Anstieg von $35 pro Kopf für 30% der Untersuchungsgesamtheit gemeint, also `$1.05m`.

Die Standardbeschriftungen für die vertikale Achse für ein stetiges Ergebnis beinhalten (photp), was für pro Kopf der Untersuchungsgesamtheit steht.

Qini-Kurve als Diagnose

Beim Bewerten und Vergleichen der Modellleistung wird dringend empfohlen die Qini-Kurve, als auch den Wert q_0 zu überprüfen. Eigenschaften robuster Modelle beinhalten:

- *Gleich bleibend oder eingipflig.* Idealerweise sollte die Qini-Kurve zu einer einzigen Spitze ansteigen (oder zu einem einzigen Tiefpunkt abfallen). Falls diese Spitze (oder Tiefpunkt) nicht bei 100% liegt, sollte sie danach auf den 100%-Wert bei 100% ansteigen (oder abfallen). Obwohl kleinere Schwankungen akzeptabel sind, weisen größere Schwankungen für gewöhnlich auf ein instabiles Modell hin.
- *Nah beieinander liegende Trainings- und Validierungskurven.* Natürlich ist das Ziel, dass die Qini-Kurven für die Trainings- und Validierungssätze nah beieinander liegen. Bei der Auswahl zwischen zwei Modellen ist es normalerweise sicherer das Modell mit der größeren Übereinstimmung zwischen Validierungs- und Trainingsleistung anstelle des Modells mit der besseren absoluten Leistung zu wählen.

  Ungeachtet dieser allgemeinen Empfehlung wird die Validierungsleistung beim Modellieren mit Portrait Uplift in der Regel weniger stark sein als jene bei konventioneller Modellierung, da es sich bei

  1. Uplift um einen zweitrangigen Effekt handelt, der nur schwer genau zu messen und noch schwieriger zu modellieren ist.
  2. Portrait Uplift wird normalerweise speziell in Situationen verwendet, in denen das Datenvolumen begrenzt ist, wodurch die Validierung weniger verlässlich wird.
  3. Validierungsdatensets sind häufig kleiner als Trainingsdatensets und damit von Natur aus in höherem Maß fehleranfällig.

- *Cutoff-Auswahl.* Im Prinzip ist der ideale Cutoff der Maximalwert (oder Minimalwert) des Diagramms, wenn das Ziel lautet, einfach die Gesamtanzahl der zunehmenden "Resonanzen" zu maximieren. Dennoch ist ein Cutoff am Maximalwert (oder Minimalwert) nicht immer die sicherste Option. Es sollte berücksichtigt werden, dass die Rangordnung der Daten, keine Trainingsdaten von Modellen, normalerweise schwächer ist als die Rangordnung der Trainingsdaten. Dementsprechend (wenn es keine negativen Effekte gibt) müssen mehr Personen öfters anvisiert werden, als grundsätzlich durch das Modell vorhergesagt wird, um den höchsten zunehmenden Uplift zu erzielen.

  Natürlich ist die Auswahl des Cutoff komplexer wenn Kosten berücksichtigt werden.

All diese Anmerkungen sind nur als Richtlinien zu verstehen, wobei die beste Vorgehensweise stark problem- und situationsabhängig ist.

Ergebnisse und Berichte

Die Ergebnisse aus einer Reihe von Modellerstellungsdurchläufen werden in einer XML-Datei gespeichert, die normalerweise den Namen `out.xml` hat. Anschließend wird die Nachbearbeitung durchgeführt, um diese XML in eine Reihe von Qini-Diagrammen in einer PDF-Datei und eine Reihe von Modellberichten als HTML-Datei zu konvertieren. Zusätzlich ist es möglich, allerdings nur bei der Variablenauswahl, einen Excel-Bericht zu erstellen.

Bei der Ausführung aus Portrait Miner werden beide automatisch erstellt und die Excel-Version wird angezeigt.

Bei der Ausführung aus der Befehlszeile werden die Berichte durch den Befehl

```
qsupliftresults
```

für einen HTML-Bericht oder dem Befehl

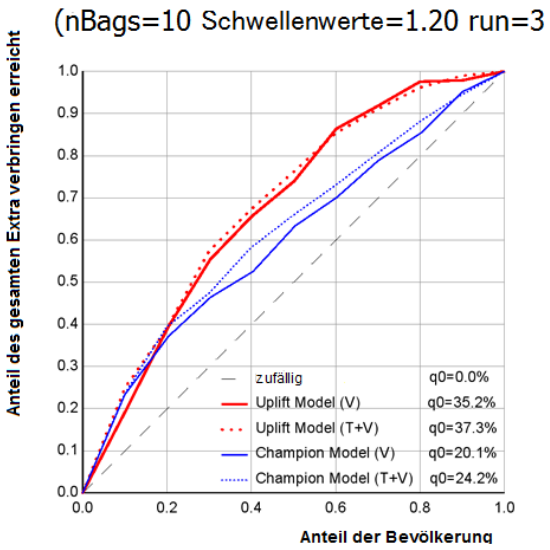
```
uogenreport
```

für einen Excel-Bericht erstellt.

Im Fall einer Excel-Berichterstattung gibt es eine Begrenzung der Anzahl der Modell- und Parametersätze, die vom Berichtsrahmen unterstützt werden. Es gibt eine Begrenzung auf 64 Parametersätze und eine Begrenzung auf 320 Modelle.

Die nachfolgende Dokumentation beschreibt die HTML-Berichterstattung. Die Excel-Berichterstattung enthält ähnliche Informationen, zeigt aber zusätzlich Finanzdiagramme und -zahlen an, die bei der Ausführung der Variablenauswahl auf den vorgegebenen ROI-Parametern basieren. Trotzdem können diese Finanzparameter (gelb markierte Zellen) "im erstellten Excel-Arbeitsblatt im Nachhinein" bearbeitet werden, wobei die Tabellen dementsprechend aktualisiert werden.

Die PDF-Datei enthält Qini-Diagramme die folgendermaßen aussehen können:



Der Modellbericht enthält eine Anzahl Tabellen, die möglicherweise einer Erklärung bedürfen.

Auswertungsprofile

Modellberichte enthalten ein Auswertungsprofil (Abbildung **8**). Dieses zeigt eine Tabelle mit allen den Feldern, die im Bagging-Baum verwendet und der Bedeutung nach absteigend sortiert wurden. Die Zahl in jedem Feld zeigt den durchschnittlich vorhergesagten Uplift ("Auswertung") an, der den Datensätzen (Kunden) in dieser Klasse für das betreffende Feld zugeordnet wurde. Die Farbcodierung zeigt mithilfe einer Legende der "Dichtekarte" das Gleiche an (die Legende wird oben im Bericht dargestellt).

In dem Beispiel ist die wichtigste Variable BehaviourSeg und die höchsten Durchschnittsauswertungen werden niedrigen und SMS-Kunden (jeweils ca. `+11.6%`) zugeteilt, während die niedrigste Durchschnittsauswertung dem Hoch-Segment (-8.94%) zugeteilt wird. Der gelbe Mittelpunkt der Skala wird auf den allgemeinen Uplift für die Untersuchungsgesamtheit festgelegt; rot steht für überdurchschnittlich und blau für unterdurchschnittlich. Kleine Klassen und Klassen mit undefinierten Uplift-Werten werden grau dargestellt.

Es ist wichtig zu beachten, dass das Auswertungsprofil den Effekt der Auswertung mithilfe des Modells beschreibt, aber das Modell nicht definiert. Der Wert in einer Zelle, z.B. die `-8.94%` im Hoch-Segment, bedeutet nicht, dass die Kundenauswertung um diesen Betrag reduziert wird, nur weil er im Hoch-Segment liegt: tatsächlich bedeutet dies, dass der durchschnittlich vorhergesagte Uplift, der allen Kunden in diesem Segment zugeordnet wird, `-8.94%` beträgt.

Zerlegungsvalidierung (der "Lila" Bericht)

Die Zerlegungsvalidierung aus der Modellerstellung ist mit dem für die Variablenauswahl identisch und wird in Abschnitt **Zerlegungsvalidierung (der "Lila" Bericht)** auf Seite 43 beschrieben.

Im Fall der Excel-Berichterstattung ist die Anzahl der Klassen pro Feld, die sowohl auf der Seite der Zerlegungsvalidierung als auch im Auswertungsprofil pro Modell angezeigt wird, auf 100 begrenzt. Hat ein Feld mehr als 100 Klassen und wird im Modell verwendet, wird für dieses spezielle Feld in beiden Berichten eine Warnmeldung angezeigt.

Aufteilungsgewichtung

Wenn Portrait Uplift die Erstellung eines Bagging-Modells abschließt, werden einige Informationen über die q_0 Qini-Leistung gemeldet, ebenso wie Informationen darüber, wie wichtig jedes verwendete Feld war, zum Beispiel:

Feldname	Bedeutung
u1	1.0833
n12	0.5833
c11	0.4167
u5	0.2917
c10	0.0417

Die Berechnung dieser Werte geschieht folgendermaßen. Für einen einzigen Baum wird einem Feld verwendet, um Knoten auf der Ebene *n* aufzuteilen. Ihnen wird eine Gewichtung von $1/2^{n-1}$ zugewiesen. Das heißt, dass die oberste Aufteilung mit 1 gewichtet und die nächste Ebene mit 0,5 gewichtet wird, usw. Felder können eine Gewichtung von mehr als eins erreichen, indem sie in der ersten und den unteren Ebenen des Baums erscheinen, da die Gewichtungen des gesamten Baums summiert werden. Bei einem Bagging-Modell mit *N* Bäumen werden die Gewichtungen einfach anteilsmäßig auf alle Modelle verteilt.

Kapitel 7

Parameterrichtlinien

In diesem Abschnitt:

- Einleitung 62
- Pruning-Schwellenwert 62
- Maximal Drei Versuche 63
- Baum-Aufteilungsebenen 63
- Mindestgrößen der Untersuchungsgesamtheit 63
- Minimale Anzahl an Knoten pro Baum 64
- Analysekandidaten und Klasseneinteilungen 64
- Anzahl Bags 65
- Anzahl Durchläufe 65
- Stratifizierungsvariable 65

Einleitung

Obwohl Portrait Uplift einen Großteil des Auswahlprozesses für das Modellieren von Uplift-Parameter von guter Qualität automatisiert hat, ist es möglich die Standardeinstellungen in einigen Anwendungen leicht zu verbessern. Es ist schwierig allgemeine Anleitungen zur Verbesserung von Parametereinstellungen auszugeben, da diese notwendigerweise unabhängig vom behandelten Problem und der Größe und Struktur der verfügbaren Daten sind. Stattdessen stellen wir einige grobe Richtlinien und Faustregeln als Startpunkt für eine eigene Untersuchung zur Verfügung.

Die Parameter mit dem größten Einfluss sind möglicherweise:

- Der Pruning-Schwellenwert
- Die maximale Anzahl von Baumversuchen
- Baum-Aufteilungsebenen
- Die Mindestgrößen der Untersuchungsgesamtheit
- Die minimale Anzahl von Knoten pro Baum
- Die Analysekandidaten und Klasseneinteilungen
- Die Anzahl von Bags (d.h. Bäume pro Bagging-Modell).
- Die Anzahl von Durchläufen (d.h. die Anzahl von zufälligen Wiederholungen bei jedem Parameterset)
- Die Stratifizierungsvariable

Diese interagieren auf offensichtlicher Weise miteinander, dennoch beschrieben wir jede kurz.

Pruning-Schwellenwert

Der Pruning-Schwellenwert steuert den Ausgleich zwischen Baumgröße (und dementsprechend einer potentiellen Genauigkeit) und Stabilität der Vorhersagen. Bestimmt der Benutzer eine minimale Baumgröße, ist es normalerweise wünschenswert, aggressiv in Richtung dieser Größe zu bereinigen.

In der Praxis ist eine gute Herangehensweise die Verwendung des niedrigsten Pruning-Schwellenwertes, der nicht zu einer unannehmbaren Anzahl von Bäumen führt, die zurückgewiesen werden, weil sie nach dem Pruning zu wenig Knoten haben. Oft ist es sinnvoll ungefähr 50 — 75% der erstellten Bäume ablehnen zu lassen.

Die von der Software bereitgestellte Information über die Standardabweichung beim Uplift ist bei der Einstellung dieses Parameters nützlich. In der Praxis sind bei binären Problemen Werte im Bereich `0.5%` bis `3.0%` am effektivsten, obwohl wir auf Probleme gestoßen sind, bei denen niedrige Werte wie `0.1%` und hohe Wert wie `5%` gut funktioniert haben.

Bei stetigen Problemen ist es schwer einen typischen numerischen Wert zu bestimmen, da er vom Umfang der Daten und der Abweichung innerhalb der Daten abhängt. Ein Weg, um ein Gefühl für die zugrunde liegende Abweichungsrate bei den Daten zu entwickeln, ist die Betrachtung der Kreuztabellen, welche die Daten auf ein Zufallsfeld und einige arbiträre Analysekandidaten herunter brechen. Den

Schwellenwert, wie hier dargestellt, weit unterhalb der natürlichen Abweichung einzustellen, ist wahrscheinlich nicht erfolgreich.

Maximal Drei Versuche

Dies ist offensichtlich mit dem Pruning-Schwellenwert verbunden. Niedrige Pruning-Schwellenwerte erfordern hierfür in der Regel höhere Werte. Obwohl der Standard 2 ist, kann dieser oft nützlicherweise auf ungefähr 5 erhöht werden, wenn das Pruning nicht zu übertrieben war.

Baum-Aufteilungsebenen

Diese liegen normalerweise im Bereich 3 — 5, wobei 4 ein gebräuchlicher Wert ist. Tiefergehende Bäume können weitere "unausgewogene" Aufteilungen erlauben, was aber meistens zu Überanpassung führt, es sei denn dies erfolgt in Kombination mit einer Mindestgröße der Untersuchungsgesamtheit, um die Gesamtanzahl der Blättern zu begrenzen.

Mindestgrößen der Untersuchungsgesamtheit

Die Mindestgröße der Untersuchungsgesamtheit beschränkt Blätterverzweigungen darauf, mindestens die bestimmte (gewichtete) Menge sowohl der Ansprech- als auch der Kontrollgruppendatensätze aufzuweisen. Daher ist es die kleinere Untersuchungsgesamtheit (in den meisten Fällen die Kontrollgruppe), die zur Beschränkung führt.

Offensichtlich interagiert die Mindestgröße der Untersuchungsgesamtheit mit der Baumtiefe und der Mindestanzahl Knoten pro Baum.

Bei einem Baum mit *n* Aufteilungsebenen (*n*+1 Knotenebenen) wird die durchschnittliche Blattuntersuchungsgesamtheit $1/2^n$ der gesamten Untersuchungsgesamtheit betragen. Es kann aber auch, bedingt durch ungleiche Aufteilungen, zu bedeutenden Abweichungen kommen. Nehmen wir an, wir haben eine Kontrolluntersuchungsgesamtheit von 60.000 mit identischer Trainings-/Validierungsaufteilung (d.h. `tvTrainProp`=50%). Wenn wir drei Aufteilungsebenen planen, bekommen wir (potentiell) 8 Blattverzweigungen und die durchschnittliche Kontrolluntersuchungsgesamtheit wird knapp unter 4 000 liegen. Um umgekehrt besonders asymmetrische Aufteilungen zu untersuchen, bei denen wir eine einzelne Klasse mit vielleicht 20% der Untersuchungsgesamtheit bis zu drei Mal aufteilen können, müssten wir die Möglichkeit von nur $1/5^3$ der Untersuchungsgesamtheit (225) an einer Blattverzweigung erlauben. Natürlich wäre diese wohl zu klein, sodass wir eventuell einen mittleren Wert wählen müssen.

Und wieder, konkrete Anleitungen für diesen Fall zu geben ist schwierig. Trotzdem legen wir Mindestgrößen der Untersuchungsgesamtheit selten auf unter 500 fest. Meistens liegen sie eher bei 1,000 — 3,000.

Normalerweise empfehlen wir, mit der automatischen Empfehlung für eine Mindestgröße durch Portrait Uplift mit dem Parameter `autoMinPopSize` (siehe Abschnitt**autoMinPopSize (optional; Standard 1)** auf Seite 73) zu beginnen, und von da an zu variieren.

Minimale Anzahl an Knoten pro Baum

Offensichtlich interagiert diese heftig mit dem Pruning-Schwellenwert als auch mit der Anzahl von Versuchen, die die Software mit jedem Pruning-Schwellenwert durchführt, bevor sie ihn erhöht.

Da wir den Durchschnitt über eine Anzahl von Bäumen berechnen, können wir häufig flachere Bäume in Portrait Uplift verwenden als bei einem Nicht-Bagging-Ansatz.

In der Praxis legen wir die minimale Anzahl von Blattverzweigungen häufig auf 4 fest, allerdings sind Werte im Bereich von 2 bis 8 durchaus üblich.

Analysekandidaten und Klasseneinteilungen

Konventionelle Methoden zur Baumerstellung beinhalten die Auswertung von Aufteilungen zwischen jedem Satz von Datenwerten für jedes Feld und verwenden alle verfügbaren Felder. Im Zusammenhang mit Portrait Uplift entstehen jedoch große Vorteile durch das Begrenzen der Anzahl der zu überprüfenden Aufteilungen.

Der Wunsch, sowohl die überprüften Variablen als auch die Anzahl an Aufteilungen pro Variable zu beschränken, entspringt dem Problem der fehlerhaften Aufteilungen, die den Problemen ähneln, die bei anderen Formen der wiederholten Versuche entstehen.

Wenn wir zahlreiche Aufteilungen für hunderte von Feldern testen, stehen die Odds, einen fehlerhaften Zusammenhang für eine Aufteilung zu erhalten, relativ hoch. Derartige fehlerhafte Aufteilungen können wesentlich stabilere und nützlichere Aufteilungen verdrängen. Obwohl durch Pruning instabile Aufteilungen entfernt werden, wird immer noch Schaden angerichtet, da der Baumerstellungsalgorithmus nach einer entfernten Aufteilung nicht erneut erstellt wird.

Aus diesem Grund empfehlen wir normalerweise die Anzahl der Felder auf eine zu bewältigende Zahl zu begrenzen (vielleicht 10 — 25). Dies ist auch der Grund, warum die Variablenauswahl in der Software enthalten ist.

Weiterhin empfehlen wir ein besonderes Augenmerk auf die Klasseneinteilungen zu legen. Im Allgemeinen sind konstante Klassengrößen vorzuziehen, da sie weniger kleine Klassen beinhalten. Und wir empfehlen Klasseneinteilungen auszuwählen, für die der Uplift über die Klasseneinteilung vergleichsweise konstant bleibt.

Anzahl Bags

Hierzu kann leichter beraten werden, da dieses Thema weniger problemabhängig ist. Wir überprüfen normalerweise Parameter mithilfe von 5-10 Bags und erstellen endgültige Modelle mit 20 Bags.

Die Anzahl Bags (Bäume pro Modell) ist der Regel unterworfen, Rückläufe zu vermindern, und es gibt auch nur minimale Vorteile jenseits von 10 Bags, und nur ganz minimale Vorteile jenseits von 20.

Anzahl Durchläufe

Normalerweise legen wir diese Einstellung beim Überprüfen von Parametern auf mindestens 5, im Prinzip macht hier aber jeder Wert zwischen 1 und 20 Sinn.

Stratifizierungsvariable

Wie im Abschnitt **Stichproben und Stratifizierung** auf Seite 46 betont, ist eine gute Stratifizierung ziemlich hilfreich. Die Stratifizierungsvariable kann vergleichsweise viele Klassen enthalten, obwohl wir von Zahlen über 100 abraten.

Portrait Uplift erfordert, dass die Stratifizierungsvariable in Klassen mit nur einem Wert pro Klasse eingeteilt wird. Falls das nicht der Fall ist, wird die Stratifizierung nicht korrekt angewendet werden. Um dieses in der Praxis sicherzustellen sollte das Feld kategorial gemacht werden. Wenn das Feld als nicht-kategorial festgelegt wurde wird Portrait Uplift eine Warnung ausgeben und selber eine kategoriale Interpretation hinzufügen. Diese Änderung wird jedoch nicht im Eingabefocus gespeichert.

Kapitel 8

Parameter-Referenz

In diesem Abschnitt:

- Einleitung 68
- Focus-Beschreibungsparameter 69
- Parameter zur Baumspezifizierung 73
- Allgemeine Parameter 74
- Parameter für die Variablenauswahl 74
- Allgemeine Durchlaufparameter 77
- Pruning-Kontrollgruppenparameter 78
- Zufallszahlenerzeugung/Wiederholungsparameter ausführen 80
- Berichts- und Ausgabeparameter 81
- ROI-Parameter 83
- Parameter nur zur Information 83
- Beispiel für eine vollständige Parameterdatei 84

Einleitung

Parameter werden für Portrait Uplift durch eine XML-Datei, deren Name in der Befehlszeile näher beschrieben wird, bereitgestellt. Eine Dateivorlage mit dem richtigen Format (`params.xml`) wird erstellt, wenn Portrait Uplift ohne eine nähere Angabe von Argumenten ausgeführt wird.

Die XML-Datei hat die folgende Struktur (siehe Abschnitt **Beispiel für eine vollständige Parameterdatei** auf Seite 84 für ein Beispiel einer vollständigen Parameterdatei):

```
<?xml version="1.0" encoding="UTF-8"?>
<params class="Globals">
    <fd class="FocusDescription">
        <!-- Focus description parameters -->
        <focusName>foci/RetentionData.ftr</focusName>
        <modelType>binary</modelType>
        ...
    </fd>
    <treeSpec class="TreeSpec">
        <!-- tree specification parameters -->
        <minPopSize>1000</minPopSize>
        <nLevels>4</nLevels>
        ...
    </treeSpec>
    <!-- general parameters -->
    <nBagsValues>[5]</nBagsValues>
    <nRuns>1</nRuns>
    ...
</params>
```

Die Reihenfolge, in der die Parameter und Parameterblöcke in der Datei abgelegt werden, spielt keine Rolle. Einige sind erforderlich, während andere optional sind. Dies wird in der unten stehenden Beschreibung der individuellen Parameterbeschreibungen beschrieben.

Benutzer sollten beachten, dass das Auslassen eines optionalen Parameters (durch das Nichteinbeziehen des Parameters in die Datei) etwas anderes ist, als ihn einzubeziehen und dann den Wert leer zu lassen. Denn ein ausgelassener Parameter übernimmt eine Standardeinstellung, während ein leerer Parameter darauf festgelegt wurde, leer zu sein.

Für die Bearbeitung der XML-Datei wird dringend empfohlen einen XML-fähigen Editor zu benutzen. Portrait empfiehlt hierfür GNU-Emacs 21,3 mit James Clark's nxml-Modus. Wird kein XML-fähiger Editor verwendet, können Benutzer einen Web-Browser wie Internet Explorer oder Firefox zur Validierung verwenden.

Im Allgemeinen verwenden binäre (yes/no oder true/false) Parameter in Portrait Uplift numerische Werte 1 (für yes/true) und 0 (für no/false). In den meisten Fällen werden alle nicht leeren Strings als true (1) und leere Strings als false (0) gewertet. Die Verwendung von 1 und 0 wird dringend empfohlen.

Focus-Beschreibungsparameter

Die folgenden Parameter erscheinen innerhalb von `<fd>` ... `</fd>` Kennzeichnungen, außer es wurde anders festgelegt. Die ebenfalls innerhalb dieser Kennzeichnungen erscheinenden Parameter `setacsFocusName` und `allowOverwrite` werden in den Abschnitten **setacsFocusName (optional)** auf Seite 74 und **allowOverwrite (optional; Standard 0)** auf Seite 75 beschrieben.

focusName (erforderlich)

Der Name des Focus, der von Portrait Uplift bei der Modellerstellung verwendet werden soll. Dies kann ein absoluter Pfad sein (beginnend mit `/` in UNIX oder ein Laufwerksbuchstabe wie `c:` in Windows), oder bezogen auf das Verzeichnis (Ordner), in dem das Programm ausgeführt wird.

Hinweis: Dieser Parameter wird ebenfalls bei der Variablenauswahl als Name für den zu schreibenden Focus verwendet, falls `<saveBestACs>` auf 1 festgelegt wurde. Windows-Pfade können mithilfe des Windows-Trennzeichens `\` angegeben werden, wobei Portrait Uplift diese immer in Schrägstriche konvertieren wird (was Windows ebenfalls erkennt).

Hinweis: Portrait Uplift verwendet immer den Standard-Subfocus im Focus.

Leerzeichen in Verzeichnis- und Dateinamen können Probleme verursachen. Das Beste ist, diese nicht zu benutzen.

Modelltyp und Ausgabefelder: Überblick

Portrait Uplift unterstützt drei Modelltypen:

- Binäres Ergebnis ("`binary`"). Dieser Modelltyp wird für Probleme, wie Absprungsquoten/Kundenbindung, Kaufwahrscheinlichkeit usw. verwendet. In diesem Fall passt das Uplift-Modell die Ergebnisänderung aus der Ansprache an, z. B. Reduzierung der Abwanderungswahrscheinlichkeit, Erhöhung der Resonanzwahrscheinlichkeit usw.
- Stetiges oder geordnetes diskretes Ergebnis ("`real`"). Dieser Modelltyp wird für Probleme verwendet, bei denen das Ziel die Stimulation der Ebene oder Größe der Kundenaktivität ist, z.B. um die Kaufrate des Kunden zu erhöhen. In diesem Fall passt das Uplift-Modell die Änderung der Ergebniswerte als ein Ergebnis der Ansprache an.
- Zweistufig ("`twostage`"). Dieser Modelltyp wird verwendet, wenn das Kampagnenziel die Erhöhung der Frequenz der Kundenaktivität (z.B. Kauffrequenz) oder des Aktivitätengrades, durch Erhöhen der Kauf- oder Verwendungsfrequenz, ist. In diesem Fall wird das Uplift-Modell zur Schätzung der Steigerung der Frequenz oder Wahrscheinlichkeit des Ergebnisses in Kombination mit einer (gelieferten) Schätzung der Ergebnisgröße (`outcomeSizeActual`) verwendet, sofern dieses eintritt.

Abhängig vom Modelltyp, der entwickelt wurde, können bis zu drei Ausgabefelder geliefert werden:

- `objectiveField` — immer erforderlich: das Ergebnis (Ziel, abhängige Variable) für die Analyse;

- `outcomeSizeActual` — wird nur bei Zweistufenmodellen zum Vergleich von Modellvorhersagen und tatsächlichen Ergebnissen verwendet. Diese ist die *tatsächliche* Größe der Wirkung für dieses Individuum, z.B. tatsächliche Ausgaben nach einer Kampagne;
- `outcomeSizeEstimate` — wird nur für Zweistufenmodelle verwendet. Diese Ergebnisgrößenschätzung wird durch Multiplikation mit der geschätzten Änderung der Ergebniswahrscheinlichkeit vom Uplift-Modell kombiniert, um die geschätzte Änderung des Erwartungswertes der Ergebnisgröße zu erstellen. Das ist:

  E(`outcomeSize`) = U(`outcomeProbability`) × `outcomeSizeEstimate`

  wobei *U* die vom Uplift-Modell produzierte Uplift-Schätzung ist. Falls keine (separate) Schätzung der Ergebnisgröße verfügbar ist, kann, sofern bekannt, die tatsächliche Ergebnisgröße verwendet werden. Zum Beispiel können in einer Kundenbindungsanwendung sowohl die Ergebnisschätzung als auch tatsächliche Werte für den jährlichen Kundenumsatz festgelegt werden. Das Ergebnis wäre eine Qini-Kurve, die den Uplift des jährlich erhaltenen Umsatzes anstelle von erhaltenen Kundenzahlen auf Grundlage der Annahme, dass künftige Umsätze den vergangenen Umsätzen ähneln, repräsentiert. (Mit anderen Worten, es wird ein triviales Modell verwendet, das vorhersagt, dass der Umsatz eines Kunden im nächsten Jahr dem diesjährigen Umsatz entspricht.)

modelType (erforderlich)

Mögliche Werte:

- `Auto`
- `Binary`
- `Real`
- `Twostage`

Die Parameter können einen `Auto`-Wert anstelle eines tatsächlichen Modelltyps verwenden. In diesem Fall wird der Modelltyp auf Grundlage der Daten und der Zielvariable während der Laufzeit entweder auf `binary` oder `real` festgelegt. Während der Laufzeit ersetzt der errechnete Typ den `Auto`-Wert in der Ausgabeparameterdatei.

objectiveField (erforderlich)

Das Feld, das das zu modellierende Ergebnis anzeigt. Dies kann entweder ein binäres Feld mit den Werten 0 und 1 oder ein numerisches Feld mit beliebigen Werten sein. Bei der Verwendung in einem zweistufigen Modell muss dieses Feld 1 sein, wenn `outcomeSizeActual` nicht-Null ist, ansonsten 0.

outcomeSizeActual (optional)

Das Feld, das die Ergebnisgröße für jeden Kunden anzeigt. Wie oben erwähnt, wird dieses nur für zweistufige Modelle verwendet.

outcomeSizeEstimate (für zweistufige Modelle erforderlich)

Das Feld, das die geschätzte Ergebnisgröße für jeden Kunden anzeigt. Wie zuvor beschrieben, wird dieses nur für zweistufige Modelle verwendet. Normalerweise wäre dies das Ergebnis eines Regressionsmodells (oder ähnlich) über die Teilmenge von Kunden mit Ergebnisgrößen nicht-Null. Das

heißt, es handelt sich um die bedingte Schätzung der Ergebnisgröße bei einem Ergebnis, das nicht-Null ist.

partitionField (erforderlich)

Dies ist ein binäres Feld mit 1 für Personen die bereits angesprochen und 0 für Personen die nicht angesprochen wurden.

Es wird dringend empfohlen, dieses Feld nicht umzukehren, da dies häufig zu ernsthafter Verwirrung führt. Falls die Daten ein umgekehrtes Zerlegungsfeld enthalten, kann ein neues Zerlegungsfeld mithilfe von FDL abgeleitet werden. Falls das vorhandene Feld `Reversed` genannt wird, geht FDL wie folgt vor:

```
1 - Reversed
if Reversed = 1 then 0 else 1
Reversed = 0
```

Wenn das Feld Stringwerte, wie `Experimentalgruppe` und `Kontrollgruppe`, enthält, wäre einer der folgenden FDL-Ausdrücke geeignet:

```
if StringPartition = "treated" then 1 else 0
StringPartition = "treated"
```

fixedTrainValidateField (optional)

Mögliche Werte:

- Name eines vorhandenen Trainings-/Validierungsfeldes
- Leer (in diesem Fall erstellt das System automatisch eine Trainings-/Validierungsaufteilung)

Die Untersuchungsgesamtheit wird in ein *Trainingsset* (T) und ein *Validierungsset* (V) aufgeteilt. Das Validierungsset wird während der Modellerstellungsphase nicht verwendet, wird aber verwendet, um Ergebnisse des Modells zu den Validierungsdaten anzuzeigen (als auch zu den Trainingsdaten).

Normalerweise errechnet das System mithilfe von stratifizierten Stichproben eine Trainings-/Validierungsaufteilung und wählt verschiedene Aufteilungen für verschiedene Modellerstellungen aus. Der Benutzer kann jedoch die Daten in Trainings- und Validierungssätze vorzerlegen. In diesem Fall gibt dieser Parameter den Namen des verwendeten Feldes vor. In diesem Feld müssen genau zwei Werte vorhanden sein — `T` für Trainingsdaten und `V` für Validierungsdaten.

Hinweis: Falls der Benutzer die Daten zerlegen will, ist es wichtig darauf zu achten, dass beide Datensets ähnliche Verteilungen aufweisen.

stratificationField (optional, wird aber empfohlen)

Das Feld, das zur Stratifizierung der Trainings- und Validierungsaufteilung verwendet wird.

Die Variablenauswahl wird die Stratifizierungsvariable automatisch erstellen, falls ihr Feldname nicht bereits im Focus enthalten ist.

Wenn `fixedTrainValidateField` leer ist, erstellt das System seine eigene Trainings-/Validierungsaufteilung. Ist dies der Fall, kann es die Zerlegung in Bezug auf ein vorhandenes Feld stratifizieren. Das bedeutet, dass der Anteil der Datensätze in den Trainings- und Validierungszerlegungen identisch ist. [2] für die verschiedenen Werte des bereitgestellten Feldes. So kann sicher gestellt werden, dass das Trainingsdatenset repräsentativ für das gesamte Datenset ist und so die Wahrscheinlichkeit einer Überanpassung von nicht-repräsentativen Daten reduziert wird.

Es wird empfohlen, dass das betreffende Feld zumindest das Feld `partitionField`, das Feld `objectiveField` und die Klassen des Feldes `outcomeSizeActual`, falls verwendet, zusammenfassen sollte. Im Idealfall sollte es auch mindestens ein Schlüssel-Segmentierungsfeld umfassen — entweder einen starken Uplift-Treiber, falls bekannt, oder ein wichtiges Feld, um zwischen den verschiedenen Kundentypen zu unterscheiden.

In Decision Studio wäre eine passende FDL-Variablenableitung hierfür beispielsweise:

```
segindex () by Partition, Objective, TopPredictor
```

Dieses Feld sollte in individuelle Werte eingeteilt werden. Normalerweise würden wir ein Minimum von vier, eher 10 bis 100 Klassen für diese Stratifizierungsvariable erwarten.

championModelField (optional)

Der Name eines oder mehrerer Felder, die die Auswertung oder die Auswertungen eines anderen ("Top")-Modells für Vergleichszwecke enthalten. Gibt es mehr als eins, müssen sie durch Leerzeichen oder Kommas getrennt werden.

Es wird häufig vorkommen, dass der Benutzer die Leistung eines Uplift-Modells mit der eines anderen Modells vergleichen möchte. In diesem Fall kann ein Top-Modell zusammen mit einer Qini-Kurve für dieses Top-Modell bereitgestellt werden.

Hinweis: Für gewöhnlich wird keines der bereitgestellten Top-Modelle auf der gleichen Trainingszerlegung wie das Uplift-Modell aufbauen. Also Vorsicht bei Vergleichen. In einigen Fällen reichen die Validierungsdaten Daten für das Top-Modell nicht aus. Dies sollte bei Vergleichen berücksichtigt werden.

userWeightingField (optional)

Ein Feld zur Gewichtung von Datensätzen für die Variablenauswahl und die Uplift-Modellerstellung.

Die Datensätze werden nicht physisch kopiert. Es handelt sich lediglich um eine Anweisung an Portrait Uplift, jeden Datensatz im Datenset so zu behandeln, als würde er mehrere Male (möglicherweise gestückelt) gleich dem Wert in diesem Feld erscheinen.

Eine derartige Fallgewichtung wird in der Regel verwendet, um die verschiedenen Verzerrungstypen, die im Modellierungsdatenset auftauchen können, auszugleichen. Im Abschnitt **Ansprechverzerrung und Modellvalidierung** auf Seite 32 werden verschiedene Szenarien beschrieben.

Hinweis: Da der Signifikanzgrad normalerweise mit der Quantität zunimmt, ist der Ansatz des Modellierens für die konstante Skalierung des Gewichtungsfeldes nicht invariant. Zum Beispiel

[2] Die Anteile sind so ähnlich wie unter den vorgegebenen Integer-Einschränkungen möglich und haben identische Erwartungswerte

könnte eine große, konstante Gewichtung einzelne Datensatzvorkommen weit signifikanter erscheinen lassen als eine Einheitengewichtung.

yQiniAxis (optional, standardmäßiger kumulativer Uplift (pc pt))

(Dieser Parameter erscheint innerhalb des allgemeinen Parameterbereichs.)

Dieser Parameter bestimmt den Text der Vertikalachse des Qini-Diagramms. Im Fall von stetigen Ergebnissen empfehlen wir das Qualifikationsmerkmal "(photp)" zu verwenden, was für "per head of total population" steht und im Abschnitt **Vertikale Achse für stetige Ergebnisse (photp)** auf Seite 55 beschrieben wird.

Parameter zur Baumspezifizierung

Diese Parameter erscheinen innerhalb der `<treeSpec>` ... `</treeSpec>`-Kennzeichnungen, außer es wurde anders festgelegt.

minPopSize (optional; Standard 2000)

Diese bestimmt die minimale Größe der Trainingsuntersuchungsgesamtheit an jedem Knoten des Baumes. Aufteilungen, die in einer kleineren Experimental- oder Kontrollgruppe als diese resultieren, werden verworfen.

Portrait Uplift kann automatisch eine Mindestgröße der Untersuchungsgesamtheit, basierend auf der allgemeinen Anstiegsraten und den verfügbaren Datenvolumina, empfehlen. Diese bieten in der Regel eine gute Ausgangsbasis. Siehe Abschnitt **autoMinPopSize (optional; Standard 1)** auf Seite 73.

autoMinPopSize (optional; Standard 1)

Dieser Parameter erscheint im Abschnitt für allgemeine Parameter.

Wenn dieser Parameter auf 1 (d.h. true) gesetzt wurde, berechnet die Software während der Laufzeit eine angemessene Mindestgröße der Untersuchungsgesamtheit, ersetzt jeden vorgegebenen Wert im Parameter `minPopSize` innerhalb des Baum-Bestimmungsabschnitts, setzt den Parameter `minLeafnodes` auf 2, und reduziert so das Risiko, dass die automatisch erstellte Mindestgröße der Untersuchungsgesamtheit keinen möglichen Baum ergibt.

Für ein binäres Ergebnis wird die Größe basierend auf der kleinsten Untersuchungsgesamtheit ausgewählt, innerhalb der die allgemeine Lift-Rate (Unterschied der Resonanzraten zwischen Experimental- und Kontrollgruppen) vertrauensvoll ermittelt werden kann.

Wenn die allgemeine Lift-Rate niedrig oder das Ergebnis stetig ist, ist die Empfehlung für die Untersuchungsgesamtheit einfach der größte angemessene Wert, durch den die erforderliche Anzahl an Blattverzweigungen erstellt werden kann.

searchMinPopSize (optional; Standard 0)

Dieser Parameter erscheint im Abschnitt für allgemeine Parameter.

Wenn dieser auf 1 (d.h. true) gesetzt wurde, berechnet die Software Bäume für mehrere Parametersätze in gleicher Weise als wenn ein Vektor von `nBags` oder `pruneThresholds` vorgegeben ist. Die aktuellen Werte der `minPopSize` betragen 100%, 75%, 50% und 25% der angegebenen oder automatisch erstellten `minPopSize` Werte.

nLevels (optional; Standard 4)

Hiermit wird die maximale Anzahl an Aufteilungsebenen im Baum festgelegt.

In Decision Studio wird die Baumtiefe für gewöhnlich in Form der Anzahl an Ebenen der *Knoten* angegeben, wohingegen hier die Anzahl der *Aufteilungs*-Ebenen bestimmt wird. Ein Baum mit *n* Aufteilungsebenen hat *n+1* Knotenebenen. Dementsprechend produziert die Einstellung auf 4 gemäß Decision Studios normaler Terminologie "Bäume mit fünf Ebenen".

Allgemeine Parameter

Die folgenden Parameter erscheinen alle innerhalb der Spitzen `<params>` ... `</params>`-Kennzeichnungen, außer es wurde anders festgelegt.

Parameter für die Variablenauswahl

Die Variablenauswahl innerhalb von Portrait Uplift verwendet dieselbe Eingabe-XML-Datei wie die Modellerstellung. Es wird nur eine Teilmenge der Hauptparameter (hauptsächlich `focusName`, `partitionField` und `objectiveField`) zusammen mit einigen speziell zugehörigen Parametern verwendet.

setacsFocusName (optional)

Der Name des als Eingabe für die Variablenauswahl zu verwendenden Focus. Falls `saveBestACs` auf 1 gesetzt wurde, wird mithilfe des in `focusName` eingegebenen Focusnamens ein Focus mit der erforderlichen Anzahl von Analysekandidaten gespeichert. Falls der Focus bereits vorhanden ist wird er überschrieben, vorausgesetzt das `allowOverwrite` auf 1 gesetzt wurde, andernfalls kann der Focus nicht gespeichert werden.

Hinweis: Die Variablenauswahl wendet exportierte Interpretationen für alle Felder, die als Analysekandidaten in diesem Focus festgelegt wurden, an, wenn unter dem durch den Parameter `focusName` bestimmten Namen gespeichert wird und löscht alle anderen

exportierten Interpretationen. Dadurch können Sie sehen, welche Felder als Analysekandidaten für die Variablenauswahl zur Verfügung stehen.

allowOverwrite (optional; Standard 0)

Falls festgelegt, ist es der Variablenauswahl erlaubt einen bestehenden Focus (Name in `focusName` enthalten) zu ersetzen. Im anderen Fall kann der Focus nicht gespeichert werden.

nBestACs (optional; Standard: 5)

Dient der Festlegung der Anzahl der auszuwählenden Analysekandidaten. (Zu allen Analysekandidaten im Eingabefocus werden Statistiken erstellt.)

saveBestACs (optional; Standard: 1)

Falls auf 1 gesetzt, wird eine neue Kopie des Focus mit diesem Set gespeichert, sobald die besten Analysekandidaten für das Modellieren eines Uplifts identifiziert wurden. Der neue Focus wird im gleichen Verzeichnis wie der ursprüngliche Focus gespeichert. Der Name ist der gleiche, jedoch um 'Best*n*ACs' ergänzt, wobei *n* die Anzahl der ausgewählten Analysekandidaten ist. (Wenn Sie Portrait Uplift von Portrait Miner aus starten, wird an den Focus die Erweiterung 'BestACs' angefügt.) Falls dieser Focus bereits vorhanden, wird eine weitere Zahl an das Ende des Focusnamen angehängt.

setacsNEvals (optional; Standard: 5)

Bei der Auswertung von Low Qini Estimates bildet die Variablenauswahl einen Mittelwert über die Anzahl von Durchläufen. Dieser Parameter steuert diese Anzahl der Durchläufe, die standardmäßig bei 5 liegt. Falls dieser Parameter auf 1 gesetzt ist, wird die Variablenauswahl vollkommen problemlos verlaufen. Ein höherer Wert führt in der Regel zu genaueren Ergebnissen, benötigt aber eine proportional längere Ausführungszeit.

In lärmintensiven Umgebungen haben wir herausgefunden, dass es mitunter erforderlich ist, `setacsNEvals` auf Werte im Bereich von 25 — 100 zu erhöhen (Weitere Informationen finden Sie unten).

setacsSigmaWeight (optional; Standard: Auto)

Bei der Low Qini Estimates-Auswertung errechnet die Variablenauswahl den durchschnittlichen Qini minus einem Vielfachen *w* von der Standardabweichung. Das heißt, der Low Qini Estimate ist:

$\mu(q_0) - w\sigma(q_0)$.

Der Parameter *w* wird durch den Parameter `setacsSigmaWeight` gesteuert. Er wird standardmäßig auf "auto" gesetzt, wobei der Wert auf $1/\sqrt{n}$ festgelegt wird und *n* die Anzahl der Durchläufe ist, die den Mittelwert des Low Qini Estimate bilden (z.B. der Parameter `setacsNEvals`). Dadurch wird der Low Qini Estimate festgelegt auf

$\mu(q_0) - \sigma(q_0)/\sqrt{n}$.

was die ungefähre Untergrenze eines Konfidenzintervalls für q_0 ist.

In Situationen, in denen die Abweichung sehr groß ist, könnte dieser standardmäßige Low Qini Estimate (mit `setacsNEvals`=5 und einem effektiven Wert für `setacsSigmaWeight` von 1/√5 □ 0,447) nicht einwandfrei arbeiten oder sogar Felder mit eindeutig positiven Low Qini Estimates produzieren. Außerdem ist die Rangordnung der Felder in diesem Fall unter Umständen nicht hilfreich. Wenn die auftritt kann es hilfreich sein, die Anzahl der Bewertungen zu erhöhen (`setacsNEvals`), wodurch das Konfidenzintervall gestrafft und infolgedessen `setacsSigmaWeight` effektiv reduziert wird, wenn es auf auto gesetzt ist.

In lärmintensiven Umgebungen haben wir herausgefunden, dass es mitunter erforderlich ist, `setacsNEvals` auf Werte im Bereich von 25 — 100 zu erhöhen.

Hinweis: Die für die erneute Klasseneinteilung der Felder benutzten Low Qini Estimates ignorieren den Wert dieses Parameters.

setacsRebin (optional; Standard: 0)

Eine detaillierte Beschreibung dieses Parameters finden Sie im Abschnitt **Automatisierte Klasseneinteilung** auf Seite 40.

- Falls `setacsRebin` auf Null gesetzt wurde, wird keine erneute Klasseneinteilung durchgeführt und es werden nur Analysekandidaten ausgewählt.
- Falls `setacsRebin` auf 1 gesetzt wurde (oder auf ein beliebiges nicht-Null-Integer), optimiert die Software die Klasseneinteilung aller nicht datumsbezogenen Analysekandidaten und schreibt eine Datei `binnings.xml` in das Ergebnisverzeichnis.
- Letztendlich, falls `setacsRebin` auf den Namen einer XML-Klasseneinteilungsdatei gesetzt wurde, legt die Software die Klasseneinteilungen auf Felder fest, so wie in der Datei angegeben.

setacsRebinCriterion (optional; Standard: T)

Damit wird die Maßeinheit für Auswahl der Klasseneinteilung gesteuert. Mögliche Werte sind

- `L`, für den höchsten Low Qini Estimate;
- `M`, für den höchsten Mean-Qini;
- `S`, für den niedrigsten Standardfehler;
- `T`, für den höchste Turning-Point Adjusted Qini (TPAQ).

TPAQ wird nur für numerische Felder definiert, für kategoriale Felder wird `S` verwendet, sofern dieser Parameter auf `T` gesetzt wurde. (Für Wertedefinitionen siehe Abschnitt **Optimierte Klasseneinteilungen** auf Seite 40.)

setacsRebinMean (optional; Standard: Trim)

Damit wird gesteuert, wie die Punkte, die zur Auswertung eines niedrigen Qini für eine optimierte Klasseneinteilung verwendet werden, zu behandeln sind. Falls auf `Normal` gesetzt, werden alle Punkte behalten. Falls auf `Trim` gesetzt werden die höchsten und niedrigsten Qini-Werte verworfen. Falls auf `Winsorize` gesetzt werden die höchsten und niedrigsten Qini-Werte durch die zweithöchsten und zweitniedrigsten ersetzt.

setacsUseRebinLQE (optional; Standard: 1)

Falls auf 1 gesetzt, anstatt Low Qini Estimates neu zu berechnen, werden Werte für die Variablenauswahl verwendet, die durch eine optimierte Klasseneinteilung berechnet wurden.

Dieser Wert wird ignoriert, falls die optimierte Klasseneinteilung nicht angewendet wird.

Während der Standardwert 1 eine höhere Geschwindigkeit ermöglicht, ist er bei Problemen wegen hohem Lärm weniger genau, sodass Sie in diesen Fällen den Wert besser auf Null (0) setzen.

Allgemeine Durchlaufparameter

Es gibt drei Hauptparameter um die Anzahl der zu erstellenden Modelle zu steuern. Diese bestimmen die Anzahl an zu verwendenden Bags, eine Reihe zu verwendender Schwellenwerte und eine Anzahl von Durchläufen, die bei jeder Einstellung ausgeführt werden.

In Pseudo-Code besagt die Hauptschleife:

```
for b in nBagsValues:
    for t in pruneThresholdValues:
        for i = 1, 2, ..., nRuns:
           Build Uplift Model
```

Falls also die Anzahl der verwendenden Bags 10 und 20 und die zu verwendenden Pruning-Schwellenwerte `1.2, 1.1` und `1.0` betragen, und 5 Durchläufe pro Wert ausgeführt werden sollen, werden 30 Uplift-Modelle wie folgt erstellt:

nBags	pruneThreshold	nRuns
10	1.2	5
10	1.1	5
10	1.0	5
20	1.2	5
20	1.1	5
20	1.0	5

nBagsValues (erforderlich)

Form: `<nBagsValues>[10,20]<nBagsValues/>`

Werteset für nBags — die Anzahl von Bags, die bei der Erstellung eines Bagging-Uplift-Modells verwendet werden soll.

Leitfaden: Normaler Bereich ist 5 bis 20.

Selbst wenn nur ein einziger Wert für `nBags` festgelegt ist, muss er in eckigen Klammern stehen. Um genau 10 Bags für alle Durchläufe zu verwenden, benutzen Sie `<nBagsValues>[10]<nBagsValues/>`. Wenn mehrere Werte festgelegt sind, müssen diese durch Kommas voneinander getrennt werden.

pruneThresholdValues (erforderlich)

Form: `<pruneThresholdValues>[1.2%, 1.1%, 1.0%]<pruneThresholdValues/>`

Festlegen der Pruning-Schwellenwerte. Angenommen, die `pruneMethod` ist auf `sd` festgelegt, dann wird dadurch die maximal erlaubte Standardabweichung des Uplifts an jedem Knoten (nicht Wurzel) im Baum über acht neue Stichproben-Zerlegungen der Trainingsdaten gesteuert.

Selbst wenn nur ein einziger Wert für `pruneThresholdValues` festgelegt ist, muss er in eckigen Klammern stehen. Um also den Schwellenwert von 1% für alle Durchläufe zu verwenden, benutzen Sie

```
<pruneThresholdValues>[1.0%]<pruneThresholdValues/>
```

Wenn mehrere Werte festgelegt sind, müssen diese durch Kommas voneinander getrennt werden.

Für binäre und zweistufige Modelle wird dies normalerweise als Prozentsatz einschließlich eines expliziten Prozentzeichens (%) ausgedrückt. Die Eingabewerte [`1.2%, 1.1%, 1.0%`] und [`0.012, 0.011, 0.01`] sind gleichwertig. Für reelle Modelle ist der Wert eine absolute Standardabweichung des Uplifts.

Falls `prune` auf 0 festgelegt wurde, wird dieser Wert ignoriert, trotzdem muss irgendein Wert bereitgestellt werden.

nRuns (optional; Standard: 1)

Anzahl der Durchläufe, die für jeden Wert von `nBagsValues` und `pruneThresholdValues` durchgeführt werden sollen.

Pruning-Kontrollgruppenparameter

prune (optional; Standard: 1)

1 wenn Pruning erforderlich ist, 0 (oder leer) falls nicht. Alle anderen Pruning-Parameter werden ignoriert, wenn Pruning ausgeschaltet ist.

autoPrune (optional; Standard: 0)

Wenn dieser auf 1 (also zutreffend) gesetzt wurde, wird während Laufzeit automatisch ein einzelner Wert in `pruneThresholdValues` festgelegt.

Der automatisch erstellte Wert basiert auf der Abweichung des Uplifts, der durch eine Reihe von *nm* Zufallsstichproben mit der Größe *N/m* im gesamten Datenset vorkommt, wobei *N* die Anzahl der Datensätze im Datenset ist und *m* und *n* entsprechend auf 10 und 5 festgelegt sind.

pruneMethod (optional; Standard: sd)

Derzeit ist die einzige unterstützte Pruning-Methode das auf Standardabweichung basierende Pruning, `sd`.

minLeafNodes (optional; Standard: 4)

Bäume mit weniger als der angegebenen Anzahl von Blätterverzweigungen nach dem Pruning werden zurückgewiesen und eine neue Baumerstellung wird gestartet.

maxTreeAttempts (optional; Standard: 2)

Dies ist die maximale Anzahl an Versuchen für die Baumerstellung mit mindestens `minLeafNodes` *am festgelegten Pruning-Schwellenwert.* Falls viele Versuche nur Bäume mit zu wenig Knoten ergeben wird der Schwellenwert (für diese BaumErstellung) erhöht, indem dieser mit 1.2 multipliziert wird. Danach wird, jedes Mal, wenn eine Baumerstellung fehlschlägt (wegen zu wenig Knoten), der Schwellenwert erneut durch Multiplizieren erhöht, das heißt es gibt einen Versuch bei jedem höheren Schwellenwert. Dies wird bis zu einem Maximum von insgesamt 12 Baumerstellungen wiederholt. Wenn der Baum nach 12 Erstellungen immer noch zu wenig Knoten aufweist, wird dieser dennoch akzeptiert.

Die Software gibt, immer wenn sie einen Baum ablehnt, eine Warnung aus.

propAnalysisCandidates (optional; Standard: 1.0)

Wenn dieser auf einen Bruch $p < 1.0$ eingestellt ist, wird ein zufälliger Anteil *p* der Analysekandidaten für die Verwendung bei jeder Modellerstellung ausgewählt. Dementsprechend werden unterschiedliche Bäume (in einem einzigen Bagging-Modell) im Allgemeinen unterschiedliche Analysekandidatenfelder verwenden.

Hinweis: Es ist keine besondere Maßnahme erforderlich, um sicherzustellen, dass die Anzahl der Analysekandidaten nach diesem Vorgang größer Null ist. Portrait Uplift wird fehlschlagen, wenn ein kleiner Wert von `propAnalysisCandidates` verwendet wird. Dies wird dazu führen, dass keine Felder als Analysekandidaten ausgewählt werden.

Hinweis: Sie können den Wert als Prozentsatz angeben, zum Beispiel sind `1.0` und 100% für diesen Parameter gleichwertig.

Zufallszahlenerzeugung/Wiederholungsparameter ausführen

Portrait Uplift verwaltet Zufallszahlen mit Sorgfalt, um das Reproduzieren eines ganzen Durchlaufs oder eines einzelnen Modells aus einem Durchlauf zu ermöglichen.

Wann immer Portrait Uplift ausgeführt wird, wird eine XML-Datei mit den Parametern des Durchlaufs (normalerweise `params.xml`) erstellt. Dabei ist `prngSetState` leer und `prngInitialState` wird auf den ursprünglichen Startwert des Zufallszahlengenerators für die Gruppe der Durchläufe gesetzt.

Zusätzlich zu den Ergebnissen der Durchläufe (normalerweise `out.xml`) enthält die XML-Datei eine Kopie der Parameter dieses speziellen Uplift-Modells einschließlich `prngCurrentState`.

Der Wert `prngSetState` bestimmt, wie der Zufallszahlengenerator gestartet wird und ob entweder einer oder beide `prng` Parameter verwendet werden.

prngSetState (optional; standardmäßig leer)

Die gültigen Werte sind:

- Leer. In diesem Fall werden beide Werte von `prngInitialState` und `prngCurrentState` in der Eingabedatei ignoriert und es wird ein neuer Startparameter per Zufallsprinzip erstellt. Das bedeutet, dass dieselbe Parameterdatei wiederholt verwendet werden kann um verschiedene Durchläufe zu erzeugen. (Beachten Sie jedoch, dass die Ausgabedatei `params.xml` immer die aktuell verwendeten Werte widerspiegelt.)
- `initial` (d.h. `<prngSetState>initial<prngSetState/>`). In diesem Fall wird der Wert in `prngInitialState` verwendet um den Zufallszahlengenerator zu starten.
- `current` (d.h. `<prngSetState>current<prngSetState/>`). In diesem Fall wird der Wert in `prngCurrentState` verwendet um den Zufallszahlengenerator zu starten.

Um einen individuellen Durchlauf zu reproduzieren kopieren Sie den XML-Code aus der Ausgabedatei (normalerweise `out.xml`) in eine Parameterdatei (normalerweise `params.xml`). Legen Sie `prngSetState` auf `current`, `nBags` auf den Wert `currentNBags` (in eckigen Klammern) und `pruneThresholdValues`auf den Wert `currentPruneThreshold` (in eckigen Klammern) fest. Obwohl es sich bei diesen zwei letzten Werten nicht um Eingabeparameter handelt werden Sie in der XML-Ausgabe protokolliert.

prngCurrentState (optional)

Wenn `prngSetState` auf `current` festgelegt ist, wird dieser Wert verwendet um den Zufallszahlengenerator zu starten.

Hinweis: Dieser sollte immer nur auf einen Wert festgelegt werden, der direkt aus einer XML-Ausgabe- oder Parameterdatei kopiert wurde. Normalerweise hat er eine Form ähnlich `(1, (26494, 3705, 301), None)`.

prngInitialState (optional)

Wenn `prngSetState` auf `initial` gesetzt wurde, wird dieser Wert verwendet um den Zufallszahlengenerator zu starten.

Hinweis: Dieser sollte immer nur auf einen Wert festgelegt werden, der direkt aus einer XML-Ausgabe- oder Parameterdatei kopiert wurde. Normalerweise hat er eine Form ähnlich `(1, (26494, 3705, 301), None)`.

tvTrainProp (optional; Standard: 50%)

Wenn der Parameter `fixedTrainValidateField` in den Dateibeschreibungsparametern leer ist wird eine Trainings-/Validierungsaufteilung erstellt. Dieser Parameter gibt den Anteil an Datensätzen an, der der Trainingszerlegung zugewiesen werden soll. Es muss sich dabei um eine Zahl zwischen 0 und 1.0 oder zwischen 0% und 100% handeln.

Wenn der Trainingsanteil auf 100% (oder sehr nah dran) gesetzt wurde, erstellt Portrait Uplift keine Ausgabe für ein Validierungsset oder ein kombiniertes Trainings- und Validierungsset, d.h. er wird bei `out.xml` und allen erstellten Diagrammen fehlen.

Berichts- und Ausgabeparameter

Die Standardausgabe der Modellerstellung wird in der Verzeichnisstruktur, die in Abbildung **9** dargestellt wird, angeordnet.

resultsDirStem (optional; Standard: Ergebnisse)

Der Basisname des Verzeichnissen, in dem die Ausgabedateien gespeichert werden. Falls `addTimeStamp` festgelegt wurde oder das entsprechende Verzeichnis bereits vorhanden ist, wird ein Zeitstempel hinzugefügt. Es kann sich um einen relativen oder absoluten Pfad handeln.

addTimeStamp (optional; Standard: 1)

Falls nicht Null, wird dem Ergebnisverzeichnisstamm ein Zeitstempel der Form JJJJMMTT-HHMMSS hinzugefügt.

Wenn der Ergebnisstamm auf seine Standardeinstellung (`Ergebnisse`) gesetzt wird und ein Durchlauf exakt um 18:35 Uhr am 8. September 2012 gestartet wird, wird das Ergebnisverzeichnis `ergebnisse20120908-183500` heißen.

Falls das in `resultsDirStem` bestimmte Verzeichnis bereits vorhanden ist, wird nach einem Unterstrich ein Zeitstempel hinzugefügt, selbst wenn dieser Parameter auf Null festgelegt wurde.

saveFocusStem (optional; Standard: (leer))

Wenn nicht leer, wird nach jedem Erstellen eines Modells der Focus mit dem Stammnamen in einem Unterverzeichnis des Ergebnisverzeichnisses namens `foci` gespeichert. Dem Stamm ist eine Signatur angefügt, die die Anzahl Bags, den Pruning-Schwellenwert und die Anzahl Durchläufe (beginnend mit 1) angibt.

Falls leer, werden Foci nicht gespeichert.

Hinweis: Die Software speichert unabhängig vom Status dieses Parameters immer die Metadaten für die Zwischenfoci als `.qsfm`-Dateien. Das bedeutet, dass die Zwischenfoci immer mit Hilfe von `qsimportmetadata` erstellt werden können. Dieser Vorgang kann jedoch sehr lange dauern. Wenn Sie also wissen, dass Sie sie benötigen werden, ist das Speichern der Zwischenfoci schneller.

saveModelStem (optional; Standard: Modell)

Wenn nicht leer, wird nach jedem Erstellen das Modell mit dem entsprechenden Stammnamen mit Hilfe einer XML-Darstellung von FDL in einem Unterverzeichnis des Ergebnisverzeichnisses namens `models` gespeichert. Dem Stamm ist eine Signatur angefügt, die die Anzahl Bags, den Pruning-Schwellenwert und die Anzahl Durchläufe (beginnend mit 1) angibt.

Ein normal gespeichertes Modell befindet sich in einem Unterverzeichnis des Ergebnisverzeichnisses namens `models` mit einem Namen wie

```
models/modelB03T3.00pcR0001.xml
```

der anzeigt, das 3 Bags (B03, war beispielsweise ein Durchschnitt von 3 Bäumen), ein Pruning-Schwellenwert von `3.00%` (`T3.00pc`) verwendet wurden und es 1 Batch-Durchlauf gab (R0001).

Wenn leer, werden Modelle nicht gespeichert.

saveTreeStem (optional; Standard: (leer))

Wenn nicht leer, wird jeder Baum nach dem Erstellen unter dem Stammnamen in einem Unterverzeichnis des Ergebnisverzeichnisses namens `trees` gespeichert. Dem Stamm wird eine Signatur angehängt, die Aufschluss über die Anzahl Bags, Pruning-Schwellenwert, Anzahl Durchläufe (beginnend mit 1) und die Modellnummer gibt (beginnend mit 1, und mit einem Suffix "Versuch" der Form A02, falls das Modell erneut erstellt wurde, nachdem durch Pruning zu viel vom Baum entfernt wurde).

Falls leer, werden Bäume nicht gespeichert.

xmlResultsStem (optional; Standard: aus)

Der Stammname der Datei, in der die Ergebnisse der Qini-Kurve als XML gespeichert sind.

tvNewFieldName (optional; Standard: TrainValidate)

Der Name des für die Trainings-/Validierungsaufteilung zu erstellenden Feldes, falls der Dateibeschreibungsparameter `fixedTrainValidateField` leer ist.

overallUpliftField (optional; Standard: UpliftScore)

Der Name des Feldes, das die endgültige Uplift-Vorhersage für jeden Kunden enthält.

ROI-Parameter

Bei der Erstellung eines Excel-Berichts und der Auswertung der empfohlenen Modellparameter werden folgende Parameter verwendet.

roiCurrencySymbol (optional; Standard: $)

Das zu verwendende Währungssymbol.

Die Währungssymbole für Euro (€) und britisches Pfund (£) müssen festgelegt werden, indem die Abkürzungen mit den drei Buchstaben EUR und GBP verwendet werden.

roiCostOfTreatment (optional; Standard: 0.00)

Die Kosten der Kampagnenansprache pro Kunde.

roiValueOfBenefit (optional; Standard: 1.00)

Der entstandene Nutzen pro reagierendem Kunden.

roiCostOfFulfillment (optional; Standard: 0.00)

Die anfallenden Kosten pro Resonanz auf ein Ansprechen.

Parameter nur zur Information

currentNBags (festgelegt während der Modellerstellung

Wird bei der Eingabe ignoriert, aber bei der Ausgabe (in der Datei durch `xmlResultsStem` angegeben) enthält dieser den Wert von `nBags` für das aktuelle Uplift-Modell.

Dieser kann in Verbindung mit `prngSetState` und `currentPruneThreshold` verwendet werden, um ein bestimmtes Modell neu zu erstellen.

currentPruneThreshold (festgelegt während der Modellerstellung)

Wird bei der Eingabe ignoriert, aber bei der Ausgabe (in der Datei durch `xmlResultsStem` angegeben) enthält dieser den Wert von `pruneThreshold` für das aktuelle Uplift-Modell.

Dieser kann in Verbindung mit `prngSetState` und `currentNBags` verwendet werden, um ein bestimmtes Modell neu zu erstellen.

applyFocusName

Wenn Uplift-Modelle auf neue Daten angewendet werden (Abschnitt **Bagging-Uplift-Modellen auf neue Daten anwenden** auf Seite 23), wird dieser Parameter auf den Namen des Focus festgelegt, auf den ein zuvor erstellter Satz an Modellen angewendet werden soll.

Beispiel für eine vollständige Parameterdatei

```
<?xml version="1.0" encoding="UTF-8"?>
<params class="Globals">

    <fd class="FocusDescription">
        <setacsFocusName>foci/RetentionData.ftr</focusName>
        <focusName>foci/RetentionDataACs.ftr</focusName>
        <modelType>binary</modelType>

        <objectiveField>AccountClosed</objectiveField>
        <outcomeSizeActual></outcomeSizeActual>
        <outcomeSizeEstimate></outcomeSizeEstimate>

        <partitionField>RetentionCampaign</partitionField>
        <fixedTrainValidateField></fixedTrainValidateField>
       <stratificationField>ObjectiveByTreatmentBySpend</stratificationField>

        <championModelField>RetentionScore</championModelField>
        <allowOverwrite>0</allowOverwrite>
    </fd>

    <treeSpec class="TreeSpec">
        <minPopSize>1000</minPopSize>
        <nLevels>4</nLevels>
    </treeSpec>

    <!-- overall run set parameters -->

    <nBagsValues>[5]</nBagsValues>
    <pruneThresholdValues>[0.7%]</pruneThresholdValues>
    <nRuns>1</nRuns>

    <!-- pruning control parameters -->

    <autoPrune>0</autoPrune>
    <prune>1</prune>
```

```
    <pruneMethod>sd</pruneMethod>
    <minLeafNodes>4</minLeafNodes>
    <maxTreeAttempts>2</maxTreeAttempts>

    <!-- minimum population size parameters -->

    <autoMinPopSize>1</autoMinPopSize>
    <searchMinPopSize>0</searchMinPopSize>

    <!-- problem interpretation parameters -->

    <yQiniAxis>Increase in Churn (pc pt)</yQiniAxis>

    <!-- setacs parameters (Automatic selection of analysis candidates -->

    <nBestACs>10</nBestACs>

    <!-- Random number generation / run repeater parameters -->

    <prngCurrentState>(1, (26494, 3705, 301), None)</prngCurrentState>
    <prngInitialState>(1, (26494, 3705, 301), None)</prngInitialState>
    <prngSetState></prngSetState>

    <tvTrainProp>50%</tvTrainProp>

    <!-- Reporting and output parameters -->

    <resultsDirStem>results</resultsDirStem>
    <addTimeStamp>1</addTimeStamp>
    <saveFocusStem>snapshot</saveFocusStem>
    <saveTreeStem>tree</saveTreeStem>
    <xmlResultsStem>out</xmlResultsStem>

    <tvNewFieldName>TrainValidate</tvNewFieldName>
    <overallUpliftField>UpliftScore</overallUpliftField>

    <!--  RoI parameters -->

    <roiCurrencySymbol>$</roiCurrencySymbol>
    <roiCostOfTreatment>0.00</roiCostOfTreatment>
    <roiValueOfBenefit>1.00</roiValueOfBenefit>
    <roiCostOfFulfillment>0.00</roiCostOfFulfillment>

    <!-- information-only parameters: these are not set by the user
  or needed on input, but are set and reported by UpliftPlus -->

    <propAnalysisCandidates>1.0</propAnalysisCandidates>
    <applyFocusName>1.0</applyFocusName>
    <currentNBags>5</currentNBags>
    <currentPruneThreshold>0.7%</currentPruneThreshold>

</params>
```

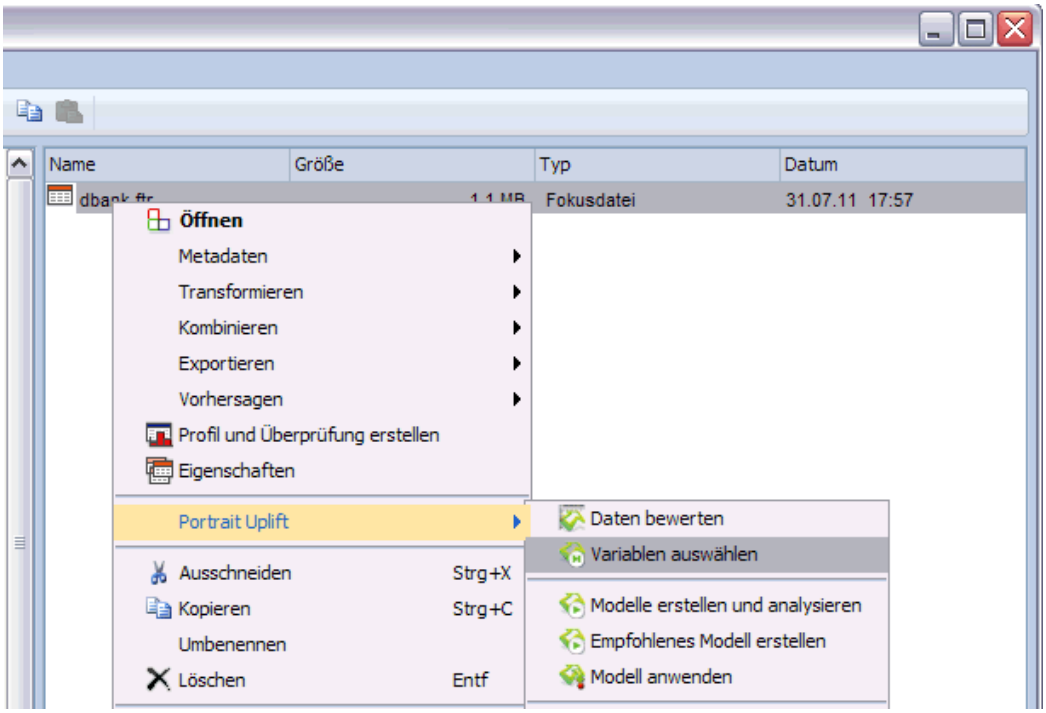


Abbildung 1: Die Option "Klicken mit der rechten Maustaste" um Variablen auszuwählen.

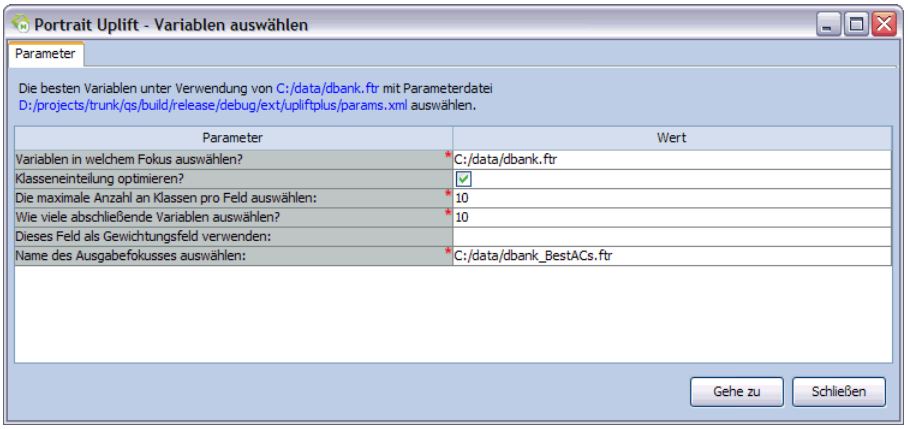


Abbildung 2: Das Dialogfeld zur Variablenauswahl.

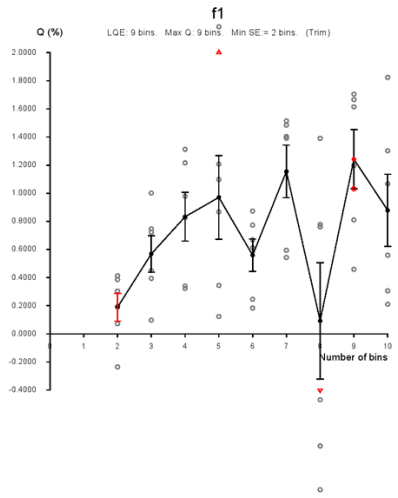


Abbildung 3: Ein Diagramm mit Qini-Werten (*Q*) als Funktion der Klassenanzahl.

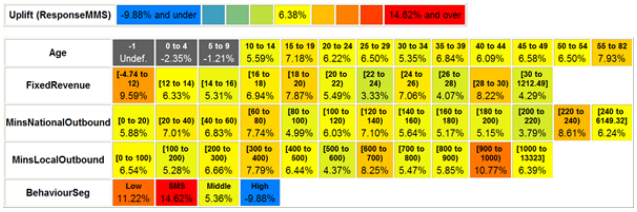


Abbildung 4: Ein Uplift Profil.

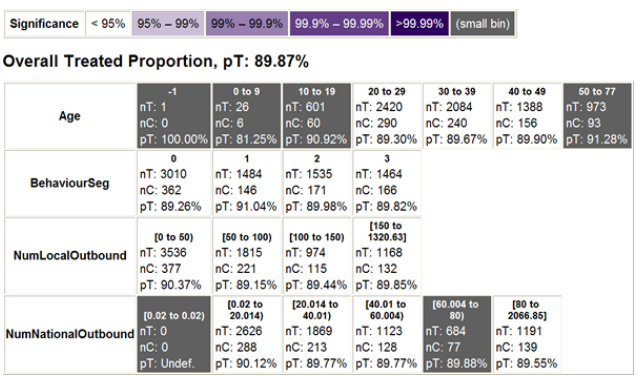


Abbildung 5: Eine Zerlegungsvalidierung. In diesem Beispiel, abgesehen von einigen grau eingefärbten Klassen mit zu wenig Datensätzen, um interessant zu sein, hat keine der Klassen einen Anteil der signifikant vom Gesamtanteil von `89.87%` abweicht.

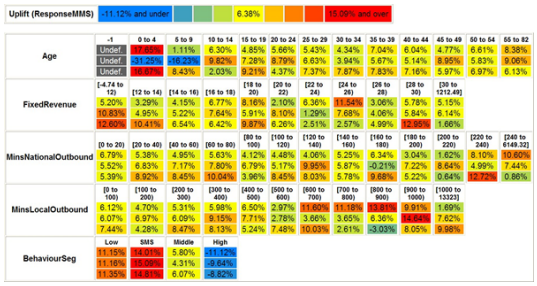


Abbildung 6: Ein Stabilitätsprofil.

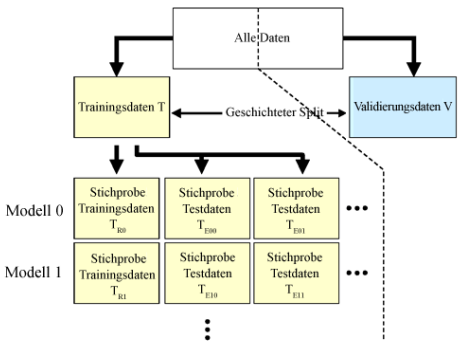


Abbildung 7: Die Struktur der Validierungs- und Trainingsdaten und neue Stichprobennahme von den Trainingsdaten, um erneut Training- und Testdatensätze zu erstellen.

Abbildung 8: Ein Auswertungsprofil.

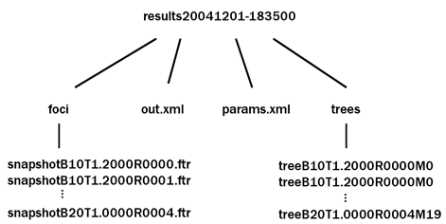


Abbildung 9: Standard-Ausgabeverzeichnisstruktur.